PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-316654
(43)Date of publication of application : 16.11.1999

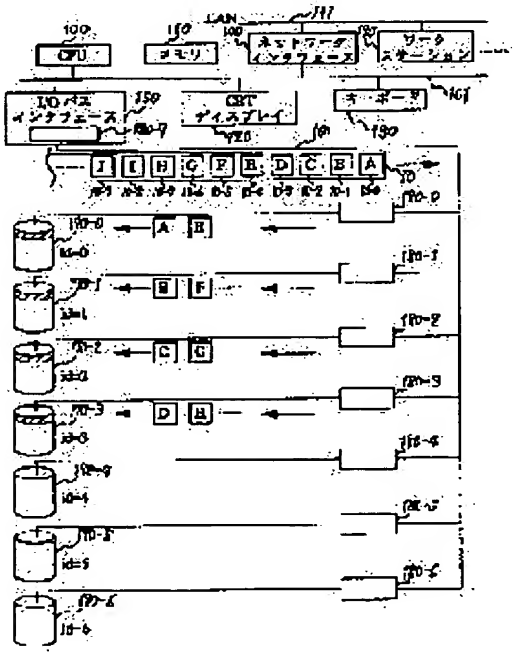
(51)Int.Cl. G06F 3/06
G06F 3/06
G11B 20/10

(21)Application number : 11-056601
(22)Date of filing : 04.03.1999

(71)Applicant : HITACHI LTD
HITACHI COMPUT ENG CORP LTD
(72)Inventor : AKISAWA MITSURU
KATO KANJI
SUZUKI HIROYOSHI
MAKI TOSHIYUKI

(54) DATA ACCESS CONTROL METHOD, COMPUTER SYSTEM AND DISK ARRAY SYSTEM

(57)Abstract:
PROBLEM TO BE SOLVED: To attain parallel operations of magnetic disk devices and also to attain a fast file access by dividing a file and performing the write and read operations to the different magnetic disk devices.
SOLUTION: In a file read mode, a file 10 stored in a memory 110 is divided and stored in four magnetic disk devices 170-0 to 170-3 (id=0 to 3) with a single data block defined as an I/O unit, for example. Then a data block A10-0 is written into the device 170-0 (id=0). When the block A10-0 is written into an internal cache of the device 170-0, the next data block B10-1 is written into the device 170-1 (id=1) without waiting for the completion of write of the block A10-0 into a disk medium from an internal cache. Thus, the next write request is issued before a write operation is carried out from an internal cache to a disk medium.



LEGAL STATUS

[Date of request for examination] 04.03.1999
[Date of sending the examiner's decision of rejection]
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
[Date of final disposal for application]
[Patent number] 3387017
[Date of registration] 10.01.2003
[Number of appeal against examiner's decision

of rejection]
[Date of requesting appeal against examiner's decision of rejection]
[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12)**公開特許公報（Ａ）**　(11)特許出願公開番号

特開平11－316654

(43)公開日　平成11年(1999)11月16日

(51)Int.Cl.[6]	識別記号	ＦＩ	
Ｇ０６Ｆ　3/06	３０２	Ｇ０６Ｆ　3/06	３０２Ｄ
	５４０		５４０
Ｇ１１Ｂ　20/10		Ｇ１１Ｂ　20/10	Ｈ

審査請求　有　　請求項の数27　ＯＬ　（全 36 頁）

(21)出願番号　特願平11－56601
(62)分割の表示　特願平4－46685の分割
(22)出願日　平成４年(1992)３月４日

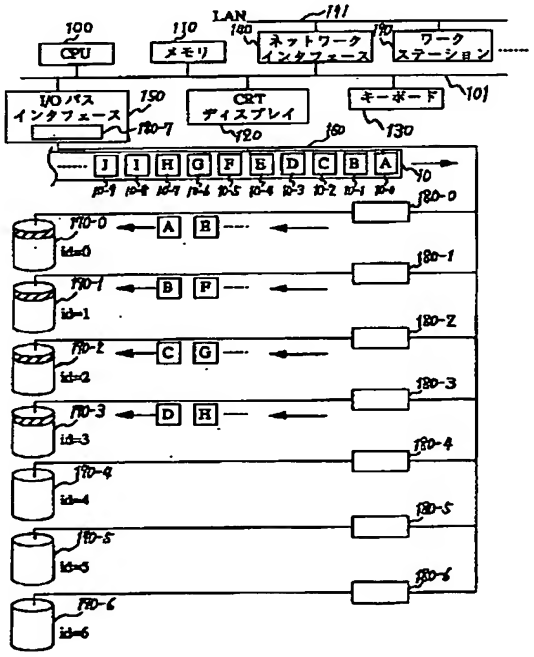
(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地
(71)出願人　000233011
日立コンピュータエンジニアリング株式会社
神奈川県秦野市堀山下１番地
(72)発明者　秋沢　充
神奈川県川崎市幸区鹿島田890番地株式会社日立製作所システム開発本部内
(74)代理人　弁理士　小川　勝男

最終頁に続く

(54)【発明の名称】　データアクセス制御方法および計算機システム並びにディスクアレイシステム

(57)【要約】
【目的】本発明の目的は、極めて低コストで高速なファイルアクセスが可能な計算機システムを提供することにある。
【構成】ディスコネクト・リコネクト機能付きＩ／Ｏバスとのインタフェースを有する計算機システム、およびＩ／Ｏバスに接続された複数の磁気ディスク装置から構成される。計算機システムは、ディスク管理情報、ファイル管理情報、およびファイル記述子対応情報を参照して、ディスクアクセスの際にファイルを分割して複数のディスクに非同期で読み書きするための制御手段を有する。
【効果】複数の磁気ディスク装置のみで他の特別な制御装置を用いずに、高速なファイルアクセスを実現することができるという効果がある。特殊なハードウェアを必要としないので、従来よりも非常に低コストで高速なファイルアクセスが可能な計算機システムを実現できる。

図1



1

【特許請求の範囲】

【請求項１】データを複数に分割した第１の分割データ群を複数の記憶装置のそれぞれに分配して格納し、上記データを複数に分割した第２の分割データ群を複数の記憶装置のそれぞれに分配して格納し、上記第１の分割データが壊れた場合は、上記壊れた第１の分割データに対応する上記第２の分割データを用いて上記第１の分割データを修復することを特徴とするデータアクセス制御方法。

【請求項２】データを複数に分割した第１の分割データ群を複数の記憶装置にそれぞれに格納し、上記データを複数に分割した第２の分割データ群を複数の記憶装置のそれぞれに格納し、上記第１の分割データをアクセスする際に、上記第１の分割データが壊れたことを検出した場合は、上記壊れた第１の分割データに対応する上記第２の分割データをアクセスすることを特徴とするデータアクセス制御方法。

【請求項３】データを複数に分割した第１の分割データ群を複数の記憶装置にそれぞれに格納し、上記データを複数に分割した第２の分割データ群を複数の記憶装置にそれぞれに格納し、新たなデータを追加する際には、上記新たなデータを複数に分割して上記第１の分割データ群を格納した複数の記憶装置のそれぞれに格納し、上記新たなデータを複数に分割して上記第２の分割データ群を格納した複数の記憶装置のそれぞれに格納することを特徴とするデータアクセス制御方法。

【請求項４】データを複数に分割した第１の分割データ群を複数の記憶装置にそれぞれに格納し、上記データを複数に分割した第２の分割データ群を複数の記憶装置にそれぞれに格納し、上記分割されたデータを削除する際には、上記第１の分割データ群を格納した複数の記憶装置に格納されている削除対象の上記分割されたデータを削除し、上記第２の分割データ群を格納した複数の記憶装置に格納されている削除対象の上記分割されたデータを削除することを特徴とするデータアクセス制御方法。

【請求項５】データを複数に分割した第１の分割データ群を複数の記憶装置にそれぞれに格納し、上記データを複数に分割した第２の分割データ群を複数の記憶装置にそれぞれに格納し、上記分割されたデータを更新する際には、上記第１の分割データ群を格納した複数の記憶装置に格納されている更新対象の上記分割されたデータを更新し、上記第２の分割データ群を格納した複数の記憶装置に格納されている更新対象の上記分割されたデータを更新することを特徴とするデータアクセス制御方法。

【請求項６】データを第１の記憶装置に格納し、上記データを第２の記憶装置に格納し、第１のデータにアクセスする際に、上記第１の記憶装置に格納した上記データが壊れたことを検出した場合、第２の記憶装置に格納している上記データをアクセスするデータアクセス制御方法において、

2

上記第１の記憶装置として複数の記憶装置が存在する場合、上記データを複数に分割し、上記複数の記憶装置のそれぞれに、分割されたデータを格納することを特徴とするデータアクセス制御方法。

【請求項７】データを第１の記憶装置に格納し、上記データを第２の記憶装置に格納し、第１のデータにアクセスする際に、上記第１の記憶装置に格納した上記データが壊れたことを検出した場合、第２の記憶装置に格納している上記データをアクセスするデータアクセス制御方法において、

上記第２の記憶装置として複数の記憶装置が存在する場合、上記データを複数に分割し、上記複数の記憶装置のそれぞれに、分割されたデータを格納することを特徴とするデータアクセス制御方法。

【請求項８】請求項６項のデータアクセス制御方法において、上記第２の記憶装置として複数の記憶装置が存在する場合、上記データを複数に分割し、上記複数の記憶装置のそれぞれに、分割されたデータを格納することを特徴とするデータアクセス制御方法。

【請求項９】請求項１項から請求項８項のデータアクセス制御方法において、上記記憶装置としてディスク装置を用いることを特徴とするデータアクセス制御方法。

【請求項１０】請求項８項のデータアクセス制御方法において、上記第１の記憶装置における複数の記憶装置は、上記第２の記憶装置における複数の記憶装置と同じ記憶装置を用いることを特徴とするデータアクセス制御方法。

【請求項１１】データを複数に分割した第１の分割データ群をそれぞれに分配して格納する複数の記憶装置と、上記データを複数に分割した第２の分割データ群をそれぞれに分配して格納する複数の記憶装置と、上記第１の分割データが破壊された場合は、上記破壊された第１の分割データに対応する上記第２の分割データを用いて上記第１の分割データを修復する制御手段とを備えたことを特徴とする計算機システム。

【請求項１２】データを格納する第１の記憶装置と、上記データを格納する第２の記憶装置と、上記データをアクセスする際に、上記第１の記憶装置に格納した上記データが壊れたことを検出した場合、第２の記憶装置に格納している上記データをアクセスする制御手段を備えた計算機システムにおいて、

上記第１の記憶装置として複数の記憶装置が存在する場合、上記データを複数に分割する分割手段と、上記複数の記憶装置のそれぞれに上記分割されたデータを格納する格納手段とを備えたことを特徴とする計算機システム。

【請求項１３】データを格納する第１の記憶装置と、上記データを格納する第２の記憶装置と、上記データをアクセスする際に、上記第１の記憶装置に格納した上記データが壊れたことを検出した場合、第２の記憶装置に格

納している上記データをアクセスする制御手段を備えた計算機システムにおいて、

上記第２の記憶装置として複数の記憶装置が存在する場合、上記データを複数に分割する分割手段と、上記複数の記憶装置のそれぞれに上記分割されたデータを格納する格納手段とを備えたことを特徴とする計算機システム。

【請求項１４】データをに格納する第１の記憶装置と、上記データを格納する第２の記憶装置と、上記第１の記憶装置に格納した上記データが壊れたことを検出した場合、第２の記憶装置に格納している上記データを用いて上記第１の記憶装置に格納した上記データを修復する制御手段を備えた計算機システムにおいて、

上記第１の記憶装置として複数の記憶装置が存在する場合、上記データを複数に分割する分割手段と、上記複数の記憶装置のそれぞれに上記分割されたデータを格納する格納手段とを備えたことを特徴とする計算機システム。

【請求項１５】データを格納する第１の記憶装置と、上記データを格納する第２の記憶装置と、上記第１の記憶装置に格納した上記データが壊れたことを検出した場合、第２の記憶装置に格納している上記データを用いて上記第１の記憶装置に格納した上記データを修復する制御手段を備えた計算機システムにおいて、

上記第２の記憶装置として複数の記憶装置が存在する場合、上記データを複数に分割する分割手段と、上記複数の記憶装置のそれぞれに上記分割されたデータを格納する格納手段とを備えたことを特徴とする計算機システム。

【請求項１６】請求項１４項の計算機システムにおいて、上記第２の記憶装置として複数の記憶装置が存在する場合、上記データを複数に分割する分割手段と、上記複数の記憶装置のそれぞれに上記分割されたデータを格納する格納手段と、上記第１の記憶装置に格納された上記壊れたデータの壊れた個所と対応する上記第２の記憶装置に格納された分割データを用いて修復する修復手段を備えたことを特徴とする計算機システム。

【請求項１７】請求項１１項から請求項１６項のディスクアレイシステムにおいて、上記記憶装置としてディスク装置を用いることを特徴とする計算機システム。

【請求項１８】データを複数に分割した第１の分割データ群をそれぞれに分配して格納する複数の記憶装置と、上記データを複数に分割した第２の分割データ群をそれぞれに分配して格納する複数の記憶装置と、上記第１の分割データが壊れた場合は、上記破壊れた第１の分割データに対応する上記第２の分割データを用いて上記第１の分割データを修復する制御手段とを備えたことを特徴とするディスクアレイシステム。

【請求項１９】データを複数に分割した第１の分割データ群をそれぞれに分配して格納する複数の記憶装置と、

上記データを複数に分割した第２の分割データ群をそれぞれに分配して格納する複数の記憶装置と、上記第１の分割データが壊れた場合は、上記壊れた第１の分割データに対応する上記第２の分割データをアクセスする制御手段とを備えたことを特徴とするディスクアレイシステム。

【請求項２０】データを複数に分割した第１の分割データ群をそれぞれに分配して格納する複数の記憶装置と、上記データを複数に分割した第２の分割データ群をそれぞれに分配して格納する複数の記憶装置と、新たなデータを追加する際には、上記新たなデータを複数に分割して上記第１の分割データ群を格納した複数の記憶装置のそれぞれに格納し、上記新たなデータを複数に分割して上記第２の分割データ群を格納した複数の記憶装置のそれぞれに格納する制御手段とを備えたことを特徴とするディスクアレイシステム。

【請求項２１】データを複数に分割した第１の分割データ群をそれぞれに分配して格納する複数の記憶装置と、上記データを複数に分割した第２の分割データ群をそれぞれに分配して格納する複数の記憶装置と、上記分割されたデータを削除する際には、上記第１の分割データ群を格納した複数の記憶装置に格納されている削除対象の上記分割されたデータを削除し、上記第２の分割データ群を格納した複数の記憶装置に格納されている削除対象の上記分割されたデータを削除する制御手段とを備えたことを特徴とするディスクアレイシステム。

【請求項２２】データを複数に分割した第１の分割データ群をそれぞれに分配して格納する複数の記憶装置と、上記データを複数に分割した第２の分割データ群をそれぞれに分配して格納する複数の記憶装置と、上記分割されたデータを更新する際には、上記第１の分割データ群を格納した複数の記憶装置に格納されている更新対象の上記分割されたデータを更新し、上記第２の分割データ群を格納した複数の記憶装置に格納されている更新対象の上記分割されたデータを更新する制御手段とを備えたことを特徴とするディスクアレイシステム。

【請求項２３】データを格納する第１の記憶装置と、上記データを格納する第２の記憶装置と、上記データをアクセスする際に、上記第１の記憶装置に格納した上記データが壊れたことを検出した場合、第２の記憶装置に格納している上記データをアクセスする制御手段を備えたディスクアレイシステムにおいて、

上記第１の記憶装置として複数の記憶装置が存在する場合、上記データを複数に分割する分割手段と、上記複数の記憶装置のそれぞれに上記分割されたデータを格納する格納手段とを備えたことを特徴とするディスクアレイシステム。

【請求項２４】データを格納する第１の記憶装置と、上記データを格納する第２の記憶装置と、上記データをアクセスする際に、上記第１の記憶装置に格納した上記デ

ータが壊れたことを検出した場合、第２の記憶装置に格納している上記データをアクセスする制御手段を備えたディスクアレイシステムにおいて、

上記第２の記憶装置として複数の記憶装置が存在する場合、上記データを複数に分割する分割手段と、上記複数の記憶装置のそれぞれに上記分割されたデータを格納する格納手段とを備えたことを特徴とするディスクアレイシステム。

【請求項２５】データをに格納する第１の記憶装置と、上記データを格納する第２の記憶装置と、上記第１の記憶装置に格納した上記データが壊れたことを検出した場合、第２の記憶装置に格納している上記データを用いて上記第１の記憶装置に格納した上記データを修復する制御手段を備えたディスクアレイシステムにおいて、

上記第１の記憶装置として複数の記憶装置が存在する場合、上記データを複数に分割する分割手段と、上記複数の記憶装置のそれぞれに上記分割されたデータを格納する格納手段とを備えたことを特徴とするディスクアレイシステム。

【請求項２６】データを格納する第１の記憶装置と、上記データを格納する第２の記憶装置と、上記第１の記憶装置に格納した上記データが壊れたことを検出した場合、第２の記憶装置に格納している上記データを用いて上記第１の記憶装置に格納した上記データを修復する制御手段を備えたディスクアレイシステムにおいて、

上記第２の記憶装置として複数の記憶装置が存在する場合、上記データを複数に分割する分割手段と、上記複数の記憶装置のそれぞれに上記分割されたデータを格納する格納手段とを備えたことを特徴とするディスクアレイシステム。

【請求項２７】請求項１８項から請求項２６項のディスクアレイシステムにおいて、上記記憶装置として磁気ディスク装置を用いることを特徴とするディスクアレイシステム。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は、ＰＣ、ワークステーションやサーバ等の計算機システム、ディスクアレイシステムに関わり、特に記憶装置に格納されたデータのデータアクセス制御方法および計算機システム並びにディスクアレイシステムに関する。

【０００２】

【従来の技術】近年のＣＰＵ性能の飛躍的な向上により、ワークステーションやサーバの演算性能は著しく向上した。しかしＣＰＵ性能の向上に比較して、Ｉ／Ｏ性能、特に磁気ディスク装置に格納されたファイルのアクセス速度の向上は十分とは言い難い。これは、磁気ディスク装置のような２次記憶装置のアクセス速度ネック、及びＩ／Ｏバスインタフェースの速度ネックが主な要因である。

【０００３】このネックを解消して高速なファイルアクセスを実現する技術の一つとして、ディスクアレイ装置（もしくはアレイディスク装置と呼ぶ）がある。これは、磁気ディスク装置（以下、簡単のためにディスクドライブあるいは単にディスクと呼ぶ）を複数台内蔵する装置で、各ディスクを並列に動作させることにより、高速なファイルアクセスを実現しようとするものである。さらに、Ｉ／Ｏバスとの接続に高速なインタフェースを用いることにより、Ｉ／Ｏバスインタフェースの速度ネックの解消も図ろうとしている。

【０００４】図４６にアレイディスク装置の構成を、また図４７にファイル読み出しの場合のタイムチャートを示す。図４６及び図４７は、各スタックが４台のディスクで構成されている例であるが、ディスク台数には増減があっても構わない。これらの図を用いてアレイディスク装置の動作を説明する。

【０００５】アレイディスク装置３００はホストＣＰＵ装置４００から読み出し命令を受けると、装置内部のコントローラ（図には示していない）が、まずスタック０のディスクｈｄ０～３にほぼ同時にデータブロック読み出しの起動をかける。スタック０の各ディスクはシーク、回転待ちのあとにディスク媒体から当該データブロックを読み出す。ディスクから読み出されたデータブロック０～３は、それぞれ別々のＦＩＦＯ（３１０～３１３）に格納される。この読み出し動作はスタック０のディスクｈｄ０～３でそれぞれ独立かつ並列に実行されるために、全体の読み出し処理の高速化が実現される。次に、これらのデータブロック０～３は時分割で装置内部の高速バス、すなわちｉ－Ｂｕｓ３２０を介してそれぞれバッファ３３０に順次シーケンシャルに転送される。このバッファ３３０で各ディスクから独立に読み出したデータブロック０～３を正しい順序に整列する。そして、高速なＳＣＳＩバス１６０を介してデータブロック０～３をホストＣＰＵ装置４００へ転送する。その後、次の読み出し命令が発行されていれば、データブロックを読み出すべきスタックのディスクに対して、アレイディスク装置内部のコントローラは直ちに読み出しの起動をかける。また、信頼性を確保するために上記データブロックにパリティを用いる方法が知られている。

【０００６】アレイディスク装置では、こうした一連の動作を繰り返すことによって、高速なファイルアクセスを実現している。

【０００７】なお、アレイディスク装置に関連する公知例としては、特開昭６４－４１０４４号「データ転送方式」、特開昭６４－２１５２５号「磁気ディスク制御装置」等がある。

【０００８】

【発明が解決しようとする課題】上述のように、従来のアレイディスク装置では、パリティを用いて信頼性を確保している。このため上記の4台のアレイディスク例で

は、例えば、ディスクが1台故障した場合、他の3台に格納されている全てのデータをアクセスし、壊れたディスクに格納していたデータの復元を行うため、時間が掛かるという問題がある。

【０００９】そこで本発明の目的は、ディスクやデータが壊れた場合、高速に修復するデータアクセス制御方法や計算機システム、ディスクアレイシステムを提供することにある。

【００１０】

【課題を解決するための手段】本発明は上記課題を解決することを目的とし、データを複数に分割した第１の分割データ群を複数の記憶装置のそれぞれに分配して格納し、上記データを複数に分割した第２の分割データ群を複数の記憶装置のそれぞれに分配して格納し、上記第１の分割データが壊れた場合は、上記壊れた第１の分割データに対応する上記第２の分割データを用いて上記第１の分割データを修復することにより上記課題を解決するものである。

【００１１】図４８に本発明の原理図を、また図４９に本発明におけるファイル読み出しのタイムチャートを示す。図４８及び図４９はディスク台数が４台の場合を示す例であるが、ディスク台数には増減があっても構わない。これらの図を用いて本発明の原理について説明する。

【００１２】本発明においては、計算機システムにＳＣＳＩバスのようなディスコネクト・リコネクト機能を有するＩ／Ｏバス１６０を設けてホストＣＰＵ装置４００と接続し、さらに、このＩ／Ｏバス１６０に内部キャッシュメモリ２０－０～２０－３を有する複数台の磁気ディスク装置１７０－０～１７０－３を接続し以下のアクセス制御を行う。

【００１３】ファイルの書き込みを行う場合には、ホストＣＰＵ装置４００は該当ファイルをサブファイルに分割して、これらを各磁気ディスク装置１７０－０～１７０－３の内部キャッシュメモリ２０－０～２０－３に次々に書き込んでゆく。ホストＣＰＵ装置４００がサブファイルを構成するデータブロックを各磁気ディスク装置の内部キャッシュメモリに書き込み終わると、各磁気ディスク装置は書き込み終了をホストＣＰＵ装置４００へ知らせる。その後、各磁気ディスク装置内部で内部キャッシュメモリに書き込まれたデータが、それぞれシークと回転待ちの後にディスク媒体に書き込まれる。当然この間、ホストＣＰＵ装置４００は他の磁気ディスク装置の内部キャッシュメモリへ他のサブファイルの書き込みを行うことができる。

【００１４】ファイルの読み出しを行う場合には、ホストＣＰＵ装置４００は、そのファイルを格納するサブファイルを読み出す命令を該当する各磁気ディスク装置１７０－０～１７０－３に対して次々に発行してゆく。発行後、Ｉ／Ｏバス１６０を介したホストＣＰＵ装置４００と各磁気ディスク装置１７０－０～１７０－３との接続をディスコネクトする。一方、各磁気ディスク装置は読み出し命令を受け付けると、それぞれシークと回転待ちの後にディスク媒体からデータを内部キャッシュメモリへ読み出して、ホストＣＰＵ装置４００に対してリコネクト要求を発行する。最も早くリコネクト要求を受け付けられた磁気ディスク装置から、ホストＣＰＵ装置４００はデータの主記憶装置への読み込みを開始する。この間、他の磁気ディスク装置もホストＣＰＵが発行した読み出し命令に基づき、ディスク媒体から内部キャッシュメモリへデータを読み出す動作を行い、読み出し完了と共にリコネクト要求をホストＣＰＵ装置４００へ発行する。このようにして読み出したサブファイルはホストＣＰＵ装置４００側で合成され、元ファイルが再生される。

【００１５】上述したファイルアクセス動作において、磁気ディスク装置１７０－０～１７０－３の内部キャッシュメモリ２０－０～２０－３へのＩ／Ｏバス１６０を介してのアクセスは高速なＩ／Ｏバスの速度で行えるため、シークや回転待ち、およびディスク媒体と内部キャッシュメモリ間のデータ転送時間に比較し、極めて短時間に行えることになる。その結果、読み出し動作も書き込み動作も、ほぼ各磁気ディスク装置がそれぞれ独立に並行して行うことになるため、システム全体としては高速なファイルアクセスが実現されることになる。

【００１６】以上に述べたように、ファイルの読み出しと書き込みのどちらの場合でも、磁気ディスク装置の内部キャッシュメモリの存在と、これに対するＩ／Ｏバスからのアクセスが高速なことを利用して、磁気ディスク装置の並列動作とＩ／Ｏバスの効率的な利用が実現できる。これにより、高速なファイルアクセスを可能とする計算機システムを提供することができる。

【００１７】上記の各機能を実現するために、本発明の提供する高速ファイルアクセス制御方法は、ディスコネクトおよびリコネクト機能を有する少なくとも１本以上の入出力バスを有し、それぞれバスインタフェース装置を介して主記憶装置に接続された該入出力バスに各々異なった装置として識別できる、内部キャッシュメモリを備えた複数の外部記憶装置が接続された計算機システムにおいて、（１）入出力バスに接続された複数の外部記憶装置の中から任意の複数の外部記憶装置を選択し、外部記憶装置グループを定義した外部記憶装置の構成情報を記憶したディスク管理情報を参照して、指定された外部記憶装置グループからこれを構成する外部記憶装置名称を求めるディスク構成情報参照ステップと、（２）アクセスの対象となる元ファイルのファイル記述子とこれを分割構成するサブファイルのファイル記述子とを対応付けて記憶したファイル記述子管理情報を参照して、指定された元ファイルのファイル記述子からこれを構成するサブファイルのファイル記述子を求めるサブファイル

記述子参照ステップと、（３）元ファイルを構成するサブファイルの、外部記憶装置グループ内の各外部記憶装置上での格納位置情報を記憶したファイル管理情報を参照して、指定されたサブファイル記述子から該サブファイルが格納されるべき、あるいはすでに格納されている該当外部記憶装置上の位置情報を求めるファイル位置情報参照ステップと、（４）上記ファイル位置情報参照ステップにより求められた各サブファイルを格納する外部記憶装置名称および該当外部記憶装置上の格納位置情報をもとに、ファイル書き込みの際には、書き込み対象となる磁気ディスク装置と当該バスインタフェース装置をコネクトして主記憶装置から該当するサブファイルを当該磁気ディスク装置の内部キャッシュメモリに転送し、転送し終えた段階で該内部キャッシュメモリからディスク媒体上への書き込み終了を待たずに、当該磁気ディスク装置をバスインタフェース装置からディスコネクトし、次の格納対象となる磁気ディスク装置をコネクトし、書き込み対象サブファイルを該磁気ディスク装置の内部キャッシュメモリへ転送し、転送終了次第ディスコネクトするという動作を次々に繰り返し、ファイル読み出しの際には、読み出し対象のサブファイルを格納している磁気ディスク装置と当該バスインタフェース装置をコネクトして、該磁気ディスク装置に対して読み出し要求を発行し、該磁気ディスク装置内のディスク媒体上のサブファイルが同装置内のキャッシュメモリへ転送されるのを待たずに、該磁気ディスク装置をディスコネクトし、直ちに次の磁気ディスク装置とコネクトして次のサブファイルの読み出し要求を発行するという動作を繰り返す一方、内部キャッシュメモリ上へサブファイルの読み込みが終了したことを伝えてきた磁気ディスク装置からリコネクトを行い、該当キャッシュメモリ上のサブファイルを主記憶上へ読み出すという動作を行うファイルアクセス制御ステップから構成される。

【００１８】なお、上記の複数台の磁気ディスク装置から構成される仮想的なディスク装置をバーチャルアレイディスク、または単にバーチャルアレイと呼ぶことにする。

【００１９】

【作用】本発明において、上記の課題を解決するための手段で述べた各処理ステップがどのように作用するのかを説明する。

【００２０】まず、処理ステップ（１）では、ファイルシステムの指定されたディレクトリにバーチャルアレイディスクを割り当て、利用可能な状態とする。この際に、ディスク管理情報を参照してバーチャルアレイディスクを構成する磁気ディスク装置を認識し、構成要素である磁気ディスク装置をバーチャルアレイディスクとして利用可能な状態にする。

【００２１】ディスク管理情報は、バーチャルアレイディスクがどの磁気ディスク装置から構成されるのかを定義する。一般に、各磁気ディスク装置は複数のパーティションに分割して使用するので、個々の磁気ディスク装置とそのパーティションを指定することで利用する領域を確定できる。ディスク管理情報はバーチャルアレイディスクが使用する磁気ディスク装置とそのパーティションを管理する。

【００２２】次に、処理ステップ（２）では、元ファイルのファイル記述子とこれを分割したサブファイルのファイル記述子との対応関係を登録したファイル記述子管理情報を参照して、サブファイルが格納された磁気ディスク装置にアクセスするためのサブファイルのファイル記述子を得る。

【００２３】ファイル記述子管理情報は、元ファイルのファイル記述子と、バーチャルアレイディスクに格納したサブファイルのファイル記述子とのあいだの対応を管理する。これにより、サブファイルの格納された複数の磁気ディスク装置をアクセスすることが可能となる。ファイル読み出し、あるいは書き込みの場合には、アプリケーションプログラムから与えられた元ファイルのファイル記述子を、ファイル記述子管理情報を参照してサブファイルのファイル記述子に変換する。また、ファイルを新規に作成して書き込みを行う場合には、新たに元ファイルとサブファイルのファイル記述子を割り当てて、ファイル記述子管理情報にこれらの対応関係を登録する。これ以降、ファイルの分割された実体であるサブファイルへのアクセスは、元ファイルのファイル記述子によって行うことができるようになる。

【００２４】次に、処理ステップ（３）では、処理ステップ（２）で得たサブファイルのファイル記述子から、サブファイルのファイル管理情報を得る。これを用いてサブファイルを格納した磁気ディスク装置へアクセスする。

【００２５】ファイル管理情報には、サブファイルが格納されている磁気ディスク装置、ストライピングブロック数、サブファイルを構成するデータブロックが格納されている位置を示す位置情報が登録されている。各サブファイルのファイル管理情報を参照して、どの磁気ディスク装置のどこに格納されたデータブロックを順にアクセスするのかを決定する。これにしたがってサブファイルを順にアクセスすることにより、元ファイルをアクセスすることとなる。

【００２６】最後に、処理ステップ（４）では、バーチャルアレイディスクに格納されたファイルをアクセスする。実際の処理では元ファイルを格納する磁気ディスクへのアクセスとなる。

【００２７】ファイルアクセスは読み出しと書き込みの場合とがあるが、いずれにせよ処理ステップ（３）で得られたサブファイルのファイル管理情報を用いる。ファイル管理情報は、サブファイルの磁気ディスク上での格納位置を管理する。したがって、各サブファイルのファ

イル管理情報からアクセスするデータブロックの格納位置を得て、次々に各サブファイルを格納する磁気ディスク装置にアクセス要求を発行する。この際に、各サブファイルをアクセスする順番はファイル記述子管理情報から得る。また、サブファイルを格納する各磁気ディスク装置に対して連続してアクセスするブロック数は、ストライピングブロックとしてファイル管理情報に登録されている。

【００２８】ファイル読み出しの場合には、読み出し対象のサブファイルが格納されている磁気ディスク装置とＩ／Ｏバスインタフェース装置とをコネクトし、読み出し要求を発行する。要求が発行されると、磁気ディスク装置はヘッドの位置決めを行い、内部キャッシュメモリへディスク媒体からデータを転送する。このために待ち時間が生じる。したがって、磁気ディスク装置とＩ／Ｏバスインタフェース装置とを一旦ディスコネクトする。この間に他の磁気ディスク装置とＩ／Ｏバスインタフェース装置とをコネクトし、読み出し要求を発行することができる。この場合にも、やはりヘッドの位置決めと内部キャッシュメモリへのデータ転送を行うための待ち時間が生じるので、さらにまた別の磁気ディスク装置に読み出し要求を発行することができる。ヘッドの位置決めが完了して読み出しが可能となった磁気ディスク装置からデータブロックを読み出し、その直後に次に読み出すデータブロックの読み出し要求を発行し、磁気ディスク装置とＩ／Ｏバスインタフェース装置とをディスコネクトするよう制御する。この様にして次々に読み出し要求を各磁気ディスク装置に発行して、サブファイルを主記憶上に読み込むことにより、元ファイルの読み込み処理を行う。

【００２９】ファイル書き込みの場合には、格納対象となる磁気ディスク装置とＩ／Ｏバスインタフェース装置とをコネクトし、書き込み要求を発行する。要求発行後にデータブロックを磁気ディスク装置へ転送する。転送されたデータブロックは磁気ディスク装置の内部キャッシュメモリに書き込まれ、ヘッドの位置決めが完了するとディスク媒体へ書き込まれる。この際に、内部キャッシュメモリへのデータブロック書き込みの終了をもって、磁気ディスク装置とＩ／Ｏバスインタフェース装置とをディスコネクトし、磁気ディスク装置への書き込みを終了とする。したがって、内部キャッシュメモリからディスク媒体への書き込み終了を待たずに、次の磁気ディスク装置とＩ／Ｏバスインタフェース装置とをコネクトして、書き込み要求発行とデータブロックの転送を行うことができる。この様にして次々に書き込み要求を各磁気ディスク装置へ発行して、元ファイルをサブファイルに分割しつつ主記憶上から書き込むことにより、元ファイルの書き込み処理を行う。

【００３０】以上の各処理ステップによるファイルアクセスにおいて、ファイル読み出しの場合には、磁気ディスク装置のディスク媒体から内部キャッシュメモリへのデータ転送の間に、次の磁気ディスク装置に読み出し要求を発行できるために、複数の磁気ディスク装置に渡って読み出し処理の多重制御を行うことが可能となる。これにより、各磁気ディスク装置でのシークと回転待ち、および内部キャッシュメモリへのデータ転送が並列に実行される。このため各磁気ディスク装置にサブファイルの読み出し要求を発行し終わると、リコネクト要求をうけて各磁気ディスク装置からデータブロックを読み出す際には、内部キャッシュメモリには主記憶へ読み出し可能なサブファイルのデータブロックが常に存在することとなる。これらを順に読み出しながら次の読み出し要求を発行し、その後に次の読み出し可能な磁気ディスク装置の処理に移って行くように制御することで、Ｉ／Ｏバスの利用効率の高い、高速なファイル読み出しが実現可能となる。

【００３１】ファイル書き込みの場合には、磁気ディスク装置の内部キャッシュメモリからディスク媒体へのデータ転送の間に、その終了を待たずに次の磁気ディスク装置に対して書き込み要求を発行できるために、複数の磁気ディスク装置に渡って書き込み処理の多重制御を行うことが可能となる。これにより、各磁気ディスク装置でのシークと回転待ち、および内部キャッシュメモリからディスク媒体へのデータ転送が並列に実行される。このため、各磁気ディスク装置にサブファイルの書き込み要求を順に発行することにより、高速なファイル書き込みが実現可能となる。

【００３２】しかも以上のファイルアクセスにおいて、アプリケーションプログラムでは元ファイルのファイル記述子のみを意識すれば良く、サブファイルに分割して格納されていることを意識する必要がない。したがって、単体の磁気ディスク装置に格納されたファイルと同様のアプリケーションインタフェースでアクセスすることが可能である。

【００３３】上述のように、本発明では以上の各処理ステップにより複数の磁気ディスク装置を順にアクセスして並列動作させることにより、高速なファイルアクセスを実現する。

【００３４】

【実施例】図１に本発明の第１の実施例を示す。本実施例の構成は，共通なデータバス１０１に接続されたＣＰＵ１００、メモリ１１０、ＣＲＴディスプレイ１２０、キーボード１３０、ネットワークインタフェース１４０、及びＩ／Ｏバスインタフェース１５０からなる計算機システムである。

【００３５】Ｉ／Ｏバスインタフェース１５０には、Ｉ／Ｏバス１６０が接続され、このＩ／Ｏバス１６０に７台の磁気ディスク装置１７０－０～１７０－６が接続されている。磁気ディスク装置の台数は７台以外の構成も取りうる。各磁気ディスク装置１７０－０～１７０－６

とＩ／Ｏバス１６０との間、及びＩ／ＯバスインタフェースＩＦ１５０内にはディスコネクト・リコネクト装置１８０－０～１８０－７があり、磁気ディスク装置１７０－０～１７０－６がＩ／Ｏバス１６０を介してＩ／ＯバスインタフェースＩＦ１５０とデータの送受信を行わないときには、両者の電気的な接続を解放（ディスコネクト）することができる。再び電気的な接続が必要になったときには、再接続（リコネクト）することができる。これらの制御はＣＰＵと磁気ディスク装置とが協調して行う。なお、上述したディスコネクト・リコネクト装置は、磁気ディスク装置１７０－０～１７０－６の内部にディスクコントローラ（図には示していない）とともに内蔵する構成も可能である。

【００３６】ネットワークインタフェース１４０はネットワーク１９１に接続され、本ネットワーク１９１に接続された遠隔地のワークステーション１９０等からのリモートアクセスを可能とする。

【００３７】次に本実施例の動作について、まずファイルをディスクに書き込む場合を例に説明する。

【００３８】本実施例の計算機システムでは、ファイル１０は固定長のデータブロックの集合としてＯＳ（オペレーティングシステム）で管理されている。すなわち、図１に示すように、Ａ１０－０、Ｂ１０－１、Ｃ１０－２、Ｄ１０－３、Ｅ１０－４、Ｆ１０－５、Ｇ１０－６、Ｈ１０－７、Ｉ１０－８、Ｊ１０－９、．．．という一連の複数のブロックから構成される。

【００３９】従来技術によるファイル書き込みの場合には、１個の磁気ディスク装置を選択し、その特定の磁気ディスク装置に対して書き込み処理を行う。すなわち、データブロックＡ１０－０、Ｂ１０－１、Ｃ１０－２、．．．を同一磁気ディスク装置、例えば１７０－０に対して書き込むことになる。その際、データブロックはまず磁気ディスク装置１７０－０の内部キャッシュメモリ（図示せず）に書き込まれ、シークと回転待ちの後にディスク媒体（図示せず）上に書き込まれる。ＣＰＵ１００は内部キャッシュメモリ（以後、簡単のために内部キャッシュとも表記する）にデータを書き込むと、ディスク媒体への書き込みが終了するまで次の書き込み処理には進まずに待ち状態（ｗａｉｔ状態）に入る。したがって、この場合ブロックＡ１０－０のディスク媒体への書き込みが完全に終了して、初めて次のブロックＢ１０－１の書き込みを実行することが可能となる。これは、ブロックＢ１０－１を磁気ディスク装置１７０－０へ書き込もうとしても、内部キャッシュ上のデータのディスク媒体への書き込みが終了していないため、書き込みが行えないからである。

【００４０】一方これに対して、本実施例では複数の磁気ディスク装置にファイルを分割して格納する。例えば、固定長のデータブロックから構成され、メモリ１１０に格納されているファイル１０を、１データブロックをＩ／Ｏの単位として、図１のｉｄ＝０から３までの４台の磁気ディスク装置１７０－０～１７０－３へ分割して格納する。この際、まずデータブロックＡ１０－０をｉｄ＝０の磁気ディスク装置１７０－０へ書き込む。データブロックＡ１０－０は磁気ディスク装置１７０－０の内部キャッシュにまず書き込まれ、次にシークと回転待ちの後にディスク媒体上の所定位置に書き込まれる。これにより書き込み処理が終了することになる。本実施例では、内部キャッシュからディスク媒体への書き込み終了を待つことなく、磁気ディスク装置１７０－０の内部キャッシュへの書き込みが終了した時点で、次のデータブロックであるブロックＢ１０－１をｉｄ＝１の磁気ディスク装置１７０－１へ書き込むという処理を行う。ここでも同様に、データブロックＢ１０－１の内部キャッシュからディスク媒体への書き込み終了を待つことなく、次のデータブロックＣ１０－２をｉｄ＝２の磁気ディスク装置１７０－２へ書き込む処理を行う。以下同様にして、データブロックＣ１０－２の内部キャッシュからディスク媒体への書き込み終了を待つことなく、データブロックＤ１０－３をｉｄ＝３の磁気ディスク装置１７０－３へ書き込む。データブロックＤ１０－３の磁気ディスク装置１７０－３の内部キャッシュへの書き込みが終了した後、ｉｄ＝０の磁気ディスク装置１７０－０へ戻り、内部キャッシュからディスク媒体へのデータブロックＡ１０－０の書き込みが終了していることを確認した上で、ブロックＤ１０－３の書き込み終了を待つことなくブロックＥ１０－４を磁気ディスク装置１７０－０の内部キャッシュに書き込む。ｉｄ＝０の磁気ディスク装置への書き込みが終了するのに十分な時間が取れない場合には、全体の磁気ディスク装置の台数を増やせば良い。これにより、さらにファイル書き込みの性能を向上させることができる。以下、データブロックＦ１０－５、Ｇ１０－６、Ｈ１０－７、Ｉ１０－８、Ｊ１０－９、．．．についても同様の方法で、異なる磁気ディスク装置１７０－１、１７０－２、１７０－３、１７０－０、．．．と、順にデータブロックを書き込んでゆく。すべて書き込み終わると、元ファイルの内容を４分割した４個のサブファイルが別々の磁気ディスク装置１７０－０～１７０－３に格納されたことになる。

【００４１】この様に、磁気ディスク装置の内部キャッシュからディスク媒体へのデータブロックの書き込みの終了を待たずに、次の磁気ディスク装置へのデータブロックの書き込み要求を発行する機能を非同期書き込みと呼ぶ。

【００４２】非同期書き込みによりサブファイルを磁気ディスク装置に書き込んでいる時、各磁気ディスク装置１７０－０～１７０－６とＩ／ＯバスインタフェースＩＦ１５０がＩ／Ｏバス１６０を介して電気的に接続されるのは、データブロックが磁気ディスク装置の内部キャッシュへ転送される間のみである。すなわち、ホストＣＰＵ

１００から書き込み要求が発行されると、磁気ディスク装置がＩ／Ｏバスインタフェース１５０と電気的に接続され、データブロックの転送が可能となる。磁気ディスク装置の内部キャッシュへのデータブロックの書き込みが終了すると、磁気ディスク装置とＩ／Ｏバスインタフェース１５０との電気的接続が開放される。その後磁気ディスク装置内部で、内部キャッシュからディスク媒体へのデータ書き込みが独立して行われる。したがって、ホストＣＰＵ１００はこの間に異なる次の磁気ディスク装置へアクセスすることができるため、磁気ディスク装置の内部キャッシュへのデータブロックの書き込みを終了したら、直ちに次の磁気ディスク装置に対する書き込み要求を発行し、この磁気ディスク装置とＩ／Ｏバスインタフェース１５０との電気的接続を確立した後にデータブロックの転送を行うことができる。

【００４３】ファイル読み出しの場合についても、本実施例は図１のファイル書き込みの場合と同様に動作する。すなわち、ｉｄ＝０の磁気ディスク装置１７０－０から順にデータブロックを読み出していく。ホストＣＰＵ１００は読み出し命令を磁気ディスク装置１７０－０から１７０－１、１７０－２、．．．へと順に発行して行く。命令を受け取った磁気ディスク装置はシークと回転待ちを行なうことなく、直ちにディスク媒体からデータブロックを読み込める場合を除き、一旦Ｉ／Ｏバスインタフェース１５０との電気的接続を開放する。その後、シークと回転待ちを行なってから、ディスク媒体の所定位置からデータを読み込むシーケンスを開始する。したがって、ホストＣＰＵ１００は磁気ディスク装置１７０－０へ読み出し命令を発行した後、直ちに次の読み出し対象となる磁気ディスク装置１７０－１に対して読み出し命令を発行することが可能となる。

【００４４】この様に、磁気ディスク装置からのデータブロックの読み出し終了を待たずに、次の磁気ディスク装置に対してデータブロックの読み出し要求を発行する機能を非同期読み出しと呼ぶ。

【００４５】以下、同様にしてホストＣＰＵ１００は次々に１７０－２、１７０－３の磁気ディスク装置へ読み出し命令を発行する。一方、各磁気ディスク装置１７０－０～１７０－３の内部では、ディスク媒体から内部キャッシュへデータが読み出されるとホストＣＰＵ１００に対してリコネクト要求を発行し、Ｉ／Ｏバスインタフェース１５０との電気的接続を確立した後にホストＣＰＵ１００へのデータ転送を行う。これら、ディスク媒体から内部キャッシュへのデータ読み出しは、それぞれの磁気ディスク装置とも相互に独立に、かつ並列に動作することが可能である。すなわち、ファイルの読み出しの場合についても、本実施例では分割したファイルを異なる複数の磁気ディスク装置から読み出すため、磁気ディスク装置の内部キャッシュの効果的な働きにより各磁気ディスク装置の並列動作が可能となる。これにより高速なファイル読み出しが可能となる。

【００４６】以上の説明にあるＩ／Ｏバス１６０を介してのディスコネクト・リコネクト機能は、例えばＳＣＳＩインタフェース(Small Computer System Interface、ANSI×3.131-1986規格)にサポートされており、本実施例のＩ／ＯバスもＳＣＳＩインタフェースの装置を用いることにより実現が可能である。

【００４７】図２にファイル読み出し(readと記す場合もある)の場合のタイムチャートを示す。(１)に従来技術である１台の磁気ディスク装置を用いる場合を、(２)に本発明のバーチャルアレイディスクを４台の磁気ディスク装置で構成して用いる本実施例の場合をそれぞれ示す。図中のＩ／Ｏバスの軸はＩ／Ｏバス１６０のタイムチャートであり、ｉｄの軸は磁気ディスク装置のタイムチャートである。Ｉ／Ｏバス軸上では、ホストＣＰＵ１００と磁気ディスク装置１７０－０～１７０－３との間で行われる読み出しリクエスト、データ転送のタイミングを示す。各ｉｄ軸上では、ディスク媒体から内部キャッシュへのデータ転送のタイミング、および内部キャッシュからＩ／Ｏバス１６０へのデータ転送のタイミングを示す。

【００４８】まず、(１)の１台の磁気ディスク装置の場合について動作を説明する。ホストＣＰＵ１００から磁気ディスク装置へ読み出し命令が発行され起動が掛かると、磁気ディスク装置内部のコントローラでコマンド解析が行われ、磁気ディスク装置のソフトウェアオーバヘッドＴ１が生じる。その後、ディスク媒体からデータブロックを読み出すためのヘッドの位置決めが行われ、(シーク時間＋回転待ち時間)Ｔ２が生じる。この間、磁気ディスク装置はディスコネクトされた状態となる。ヘッドの位置決め終了後、磁気ディスク装置はリコネクトされ、ディスク媒体からデータブロックを内部キャッシュへ読み出しながら、同時に内部キャッシュからＩ／Ｏバス１６０へ出力する。このためデータ転送時間Ｔ３が生じる。

【００４９】一方、(２)の４台の磁気ディスク装置で構成されるバーチャルアレイディスクの場合には、ホストＣＰＵ１００からｉｄ＝０～３の４台の磁気ディスク装置へそれぞれ順に読み出し命令が発行され起動が掛かる。

【００５０】まずｉｄ＝０の磁気ディスク装置１７０－０に読み出し命令が発行されると、コマンド解析後磁気ディスク装置１７０－０はシークと回転待ちに入り、同時にディスコネクトされる。これによりホストＣＰＵ１００は次のｉｄ＝１の磁気ディスク装置１７０－１に読み出し命令を発行することができるようになる。ｉｄ＝１の磁気ディスク装置１７０－１もｉｄ＝０の磁気ディスク装置と同様に、コマンド解析後シークと回転待ちに入り同時にディスコネクトされる。このとき、ｉｄ＝０の磁気ディスク装置１７０－０とｉｄ＝１の磁気ディス

ク装置１７０－１はそれぞれ独立に並列動作していることになる。さらにホストＣＰＵ１００は次のｉｄ＝２の磁気ディスク装置１７０－２、ｉｄ＝３の磁気ディスク装置１７０－３にも同様の手順で読み出し命令を次々に発行する。これらの磁気ディスク装置も同様に、コマンド解析後シークと回転待ちに入り同時にディスコネクトされる。この時点で４台の磁気ディスク装置がそれぞれ全く独立に並列動作することになる。

【００５１】ｉｄ＝３の磁気ディスク装置１７０－３がディスコネクトされた後、ｉｄ＝０の磁気ディスク装置１７０－０がヘッドの位置決めを完了してデータの読み出しが可能となっていれば、あるいはデータがディスク媒体から読み出されて内部キャッシュに格納されていれば、ホストＣＰＵ１００に対してリコネクト要求を発行する。ホストＣＰＵ１００はリコネクト要求を受け付けると、ｉｄ＝０の磁気ディスク装置１７０－０からデータの読み込みを行う。この際、あらかじめデータが内部キャッシュに格納されていれば、ディスク媒体からの読み出し時間Ｔ３より短い時間Ｔ４（Ｔ３＞Ｔ４）で高速にデータを読み出すことができる。読み込みが終了すると、ｉｄ＝０の磁気ディスク装置１７０－０に対して次の読み出し命令を発行する。ｉｄ＝０の磁気ディスク装置１７０－０は先程と同様に、コマンド解析後シークと回転待ちに入り同時にディスコネクトされる。この時、ｉｄ＝１の磁気ディスク装置１７０－１が読み出し可能となっていれば、ｉｄ＝０の磁気ディスク装置１７０－０の場合と同様にホストＣＰＵ１００は磁気ディスク装置１７０－１からのリコネクト要求を受け付けて、データを読み込む。読み込みが終了すると、ｉｄ＝１の磁気ディスク装置１７０－１に対して次の読み出し命令を発行する。そして、本磁気ディスク装置１７０－１もコマンド解析後シークと回転待ちに入り同時にディスコネクトされる。さらにホストＣＰＵ１００は次のｉｄ＝２の磁気ディスク装置１７０－２、及びｉｄ＝３の磁気ディスク装置１７０－３に対しても同様の手順でリコネクト要求を受け付け、データ読み込み、及び次の読み出し命令の発行を行う。以後、このシーケンスが図２に示されるように繰り返される。なお、本図では(シーク時間＋回転待ち時間)Ｔ２を定数と仮定しているが、実際は各読み出し命令発行の都度異なることが多いそのような場合には、最も早くリコネクト要求を出してきた磁気ディスク装置から順にデータを読み込むということも可能である。その際に、データを読み出す磁気ディスク装置を選択する順序が前後することになるが、それ以外の上記の手順は変わらない。図２では、ｉｄ＝０の磁気ディスク装置１７０－０のみがディスク媒体から直接データ転送を行い、他のｉｄの磁気ディスク装置１７０－１～１７０－３は内部キャッシュからデータ転送を行うかたちになっている。このためデータ転送時間に差が生じている。これはすなわち、ホストＣＰＵ１００がデータを読み込むまでに内部キャッシュへデータが読み込まれている場合にはデータ転送時間は短くなり、読み込まれていない場合にはディスク媒体からの読み出しとなりデータ転送時間は短くはならないためである。

【００５２】図２からも明らかなように、本発明では常に４台の磁気ディスク装置が並列に動作しており、単位時間あたりのホストＣＰＵ１００のデータ読み込み量が実効的に磁気ディスク装置の台数分倍増するという効果が得られる。すなわち、本発明によればファイルの読み出し速度がほぼ磁気ディスク装置の台数分高速化されることになる。

【００５３】図２で示したファイル読み出し(read)の、より一般化した制御フロー(１)～(８)を図３に示す。使用する磁気ディスク装置ｎ台はあらかじめ指定されており、これらにファイルが分割して格納されているものとする。これらのファイル読み出しに必要な情報は、例えばアプリケーションプログラムから与えるものとする。また、ファイルアクセスのシーケンス、および非同期読み出しの制御はホストＣＰＵ１００が行う。

【００５４】以下、制御フローの各ステップについて説明する。

【００５５】(１)ｉｄ＝０～(ｎ－１) の磁気ディスク装置に対して読み出し要求(readリクエスト)を順次発行する。

【００５６】(２)いずれかの磁気ディスク装置が読み出し可能状態となるまで待つ。

【００５７】(３)読み出し可能な磁気ディスク装置があれば次のステップ(４)へ進み、なければステップ(２)へ戻る。

【００５８】(４)読み出し可能な磁気ディスク装置のｉｄ番号をチェックし、記憶する。

【００５９】(５)ステップ(４)でチェックしたｉｄ＝ｋの磁気ディスク装置から、データを読み出す。

【００６０】(６)読み出し終了であればフローを終了し、読み出し終了でなければ次のステップ(７)へ進む。

【００６１】(７)ｉｄ＝ｋの磁気ディスク装置からさらにデータを読み出すのであれば次のステップ(８)へ進み、もうデータを読み出さないのであればステップ(２)へ戻る。

【００６２】(８)ｉｄ＝ｋの磁気ディスク装置に対して読み出し要求(readリクエスト)を発行し、ステップ(２)へ戻る。

【００６３】以上の制御フローにより、バーチャルアレイディスクからの高速なファイル読み出しを行うことができる。

【００６４】図４にファイル書き込み(writeと記す場合もある)の場合のタイムチャートを示す。(１)に従来技術である１台の磁気ディスク装置を用いる場合を、(２)に本発明のバーチャルアレイディスクを４台の磁気ディスク装置で構成して用いる場合をそれぞれ示す。図中の

Ｉ／Ｏバスの軸、ｉｄの軸の意味はそれぞれ図２と同様である。

【００６５】まず、(１)の１台の磁気ディスク装置の場合について動作を説明する。ホストＣＰＵ１００から磁気ディスク装置へ書き込み命令が発行され起動が掛かると、磁気ディスク装置内部のコントローラでコマンド解析が行われ、磁気ディスク装置のソフトウェアオーバヘッドＴ１が生じる。この時に、ホストＣＰＵ１００は書き込むデータを磁気ディスク装置へ転送し、データは直ちに内部キャッシュへ格納される。この時データ転送時間Ｔ４が生じる。これをもって、ホストＣＰＵ１００からのデータ書き込み処理は終了したものとみなされる。その後、磁気ディスク装置ではディスク媒体へデータブロックを書き込むためのヘッドの位置決めが行われ、(シーク時間＋回転待ち時間)Ｔ２が生じる。この間、磁気ディスク装置はビジー状態となる。ヘッドの位置決め後、磁気ディスク装置はデータブロックを内部キャッシュからディスク媒体へ書き込む。このためデータ転送時間Ｔ３が生じる。

【００６６】一方、(２)の４台の磁気ディスク装置で構成されるバーチャルアレイディスクの場合には、ホストＣＰＵ１００からｉｄ＝０～３の４台の磁気ディスク装置１７０－０～１７０－３へ、それぞれ順に書き込み命令が発行されて起動が掛かる。

【００６７】まずｉｄ＝０の磁気ディスク装置１７０－０に書き込み命令が発行され、引き続き書き込むデータが転送されて磁気ディスク装置の内部キャッシュへ格納される。これにより、ホストＣＰＵ１００からのデータ書き込み処理は終了したものとみなされる。磁気ディスク装置はコマンド解析後、シークと回転待ちに入る。

【００６８】ヘッドの位置決めが完了すると、内部キャッシュのデータがディスク媒体に書き込まれる。一方、ホストＣＰＵ１００は次のｉｄ＝１の磁気ディスク装置１７０－１に書き込み命令を発行し、引き続き書き込むデータを転送することができるようになる。ｉｄ＝１の磁気ディスク装置１７０－１もｉｄ＝０の磁気ディスク装置１７０－０と同様に、コマンド解析後シークと回転待ちに入り内部キャッシュからディスク媒体へのデータ書き込みを行う。このとき、ｉｄ＝０の磁気ディスク装置１７０－０とｉｄ＝１の磁気ディスク装置１７０－１は各々独立に並列動作していることになる。さらにホストＣＰＵ１００は次のｉｄ＝２の磁気ディスク装置１７０－２、ｉｄ＝３の磁気ディスク装置１７０－３にも同様の手順で書き込み命令の発行とこれに引き続くデータ転送を次々に実行する。これらの磁気ディスク装置も同様に、コマンド解析後シークと回転待ちに入り、ヘッドの位置決めが完了すると、内部キャッシュのデータをディスク媒体に書き込む。この時点で４台の磁気ディスク装置１７０－０～１７０－３が、それぞれ独立に並列動作することになる。

【００６９】ｉｄ＝３の磁気ディスク装置１７０－３に書き込み命令の発行と書き込みデータの転送を行った後、ｉｄ＝０の磁気ディスク装置１７０－０のデータ書き込みが完了していれば、次のデータの書き込みが可能である。そこで、次の書き込み命令の発行とデータ転送とを実行する。ｉｄ＝０の磁気ディスク装置１７０－０は先程と同様にコマンド解析後シークと回転待ちに入る。この時、ｉｄ＝１の磁気ディスク装置１７０－１のデータ書き込みが完了していれば、次のデータの書き込みが可能である。そこでｉｄ＝０の磁気ディスク装置１７０－０の場合と同様に、ホストＣＰＵ１００は磁気ディスク装置へ次の書き込み命令の発行とデータ転送とを実行する。そして、コマンド解析後シークと回転待ちに入る。さらにホストＣＰＵ１００は次のｉｄ＝２の磁気ディスク装置１７０－２、ｉｄ＝３の磁気ディスク装置１７０－３にも前に発行したデータ書き込みの完了を確認すると、同様の手順で次の書き込み命令の発行を行う。以後、このシーケンスが図４に示されるように繰り返される。なお、本図では(シーク時間＋回転待ち時間)Ｔ２を定数と仮定しているが、実際は各書き込み命令の発行の都度異なることが多い。そのような場合には、最も早くビジー状態が解除されて書き込み可能な状態となった磁気ディスク装置から順にデータを書き込むということも可能である。

【００７０】データを書き込む磁気ディスク装置を選択する順序が前後することになるが、それ以外の上記の手順は変わらない。

【００７１】図４からも明らかなように、本発明では常に４台の磁気ディスク装置が並列に動作しており、単位時間あたりのホストＣＰＵ１００のデータ書き込み量が実効的にディスク台数分倍増するという効果が得られる。すなわち、本発明によればファイル書き込み速度が磁気ディスク装置の台数分高速化されることになる。

【００７２】図４で示したファイル書き込み(write)の、より一般化した制御フロー(１)～(７)を図５に示す。使用する磁気ディスク装置ｎ台はあらかじめ指定されており、これらにファイルを分割して格納するものとする。これらのファイル書き込みに必要な情報は、例えばアプリケーションプログラムから与えるものとする。また、ファイルの複数磁気ディスク装置への分割、ファイルアクセスのシーケンス、および非同期書き込みの制御はホストＣＰＵ１００が行う。

【００７３】以下、制御フローの各ステップについて説明する。

【００７４】(１)ｉｄ＝０～(ｎ－１) の磁気ディスク装置に対して以下の処理を順次実行する。　１．書き込み要求(writeリクエスト)を発行する。

【００７５】２．書き込みデータを転送する。

【００７６】(２)いずれかの磁気ディスク装置が書き込み可能状態となるまで待つ。

【００７７】(3)書き込み可能な磁気ディスク装置があれば次のステップ(4)へ進み、なければステップ(2)へ戻る。

【００７８】(4)書き込み可能な磁気ディスク装置のｉｄ番号をチェックし、記憶する。

【００７９】(5)ｉｄ＝ｋの磁気ディスク装置にさらにデータを書き込むのであれば、次のステップ(6)へ進み、もうデータを書き込まないのであればステップ(2)へ戻る。

【００８０】(6)ｉｄ＝ｋの磁気ディスク装置に対して以下の処理を順次実行する。

【００８１】１．書き込み要求(writeリクエスト)を発行する。

【００８２】２．書き込みデータを転送する。

【００８３】(7)書き込み終了であればフローを終了し、書き込み終了でなければステップ(2)へ戻る。

【００８４】以上の制御フローにより、バーチャルアレイディスクへの高速なファイル書き込みを行うことができる。

【００８５】以上に説明したように本実施例ではファイルを分割して異なる複数の磁気ディスク装置への書き込み、および読み出しを行うことにより、各磁気ディスク装置の並列動作が可能となり高速なファイルアクセスが実現される。また、アレイディスク装置のように専用のハードウェアを必要とせず、例えばＳＣＳＩバスを用いることにより、７台までの磁気ディスク装置を１枚のアダプタボードで接続することが可能となる。このため、非常に低コストで高速なファイルアクセス機能を実現することができる。

【００８６】また、ＳＣＳＩインタフェースの仕様を拡張すれば、７台以上の磁気ディスク装置を接続することも可能であることは明らかである。さらに、本実施例では固定長のデータブロックを各磁気ディスク装置に分散格納する場合を例にして説明したが、可変長のデータブロックでも同様の処理が可能である。

【００８７】図６に本発明の第２の実施例を示す。第２の実施例は、ファイルが複数のサブファイルに分割されて別々の磁気ディスク装置に格納されていることをアプリケーションプログラムから意識しなくとも、ファイルにアクセスできるようにした実施例である。

【００８８】以下の各テーブルとプログラムとを用いて、バーチャルアレイディスクとして使用する磁気ディスク装置の指定、ファイルの複数サブファイルへの分割と磁気ディスク装置への対応付け、複数のサブファイルを一まとまりのファイルに見せるアクセス制御、および非同期読み出しと非同期書き込みの制御を行う。

【００８９】ディスク管理情報を格納するためのディスク管理テーブル２１０、ファイル管理情報を格納するためのファイル管理テーブル２２０、ファイル記述子対応情報を格納するためのファイル記述子対応テーブル２３０をメモリに保持する。アクセス制御プログラム２００は、バーチャルアレイディスクを対象とするファイルの書き込みと読み出しの際に、上記各テーブルを参照してアクセス制御を行う。

【００９０】上記各テーブル、およびプログラムは計算機システムのメモリに格納される。

【００９１】なお、上記の各テーブルはアプリケーションプログラムから与えることも、ＯＳの内部の管理テーブルとして管理することも、いずれの方法も可能である。

【００９２】各テーブルの具体的な構成を以下に説明する。

【００９３】図７はディスク管理テーブル２１０の説明図である。ディスク管理テーブルには、どの磁気ディスク装置のどのパーティション（分割領域）を用いてバーチャルアレイディスクを構成するのかということに関する情報が格納される。

【００９４】各磁気ディスク装置には、ＳＣＳＩインタフェースで磁気ディスク装置の識別に用いるｉｄ番号と、アプリケーションソフトウェアが磁気ディスク装置の識別に用いる磁気ディスク装置名称が割り当てられる。また、通常磁気ディスク装置は複数のパーティションに分割して使用するので、各パーティションにはパーティション番号が付けられる。したがって、ｉｄ番号または磁気ディスク装置名称とパーティション番号との組合せで、システム内の磁気ディスク装置を利用する際の単位領域の指定を行うことができる。これを“論理ディスク装置”と呼ぶことにする。本図に示した例では、ディスク装置名称ｈｄ６のディスク装置の各パーティションが論理ディスク装置として定義されており、例えば第５パーティションは名称がｈｄ６５の論理ディスク装置となる。この論理ディスク装置を複数組み合わせて、“論理バーチャルアレイディスク装置”として使用する。

【００９５】ディスク管理テーブル２１０は磁気ディスク装置のｉｄ番号または名称とパーティション番号とのマトリクスであり、これらのどの組合せの論理ディスク装置が論理バーチャルアレイディスク装置として定義されているのかを示している。

【００９６】名称がｈｄ０からｈｄ３の４台の磁気ディスク装置をバーチャルアレイディスク装置ｖａ０として定義することとし、その各第０パーティションを図１または図６に示すような論理バーチャルアレイディスク装置ｖａ００として用いる場合には、ｖａ０００からｖａ００３までを識別子として図７に示すように書き込む。これにより磁気ディスク装置ｈｄ０からｈｄ３の第０パーティションを論理バーチャルアレイディスク装置ｖａ００として定義したことになる。

【００９７】ここで識別子ｖａＸＹＺ（ただし、Ｘ、Ｙ、Ｚは一桁の数字）は、この論理ディスク装置がバー

チャルアレイディスク装置ｖａＸの構成要素であり、かつバーチャルアレイディスク装置ｖａＸの第Ｙパーティションからなる論理バーチャルアレイディスク装置ｖａＸＹの構成要素であり、さらにその論理バーチャルアレイディスク装置ｖａＸＹの第Ｚ番目の構成要素となる論理ディスク装置であるということを示している。

【００９８】なお、ディスク管理テーブル２１０の設定は、本計算機システムのシステム構成を定義する時点でシステム管理者が行う。これは、ディスク管理テーブル２１０の設定内容が、各論理ディスク装置の使用状態、すなわち通常のファイル格納に使われているのか、サブファイルの格納に使われているのかということと整合が取れていなければならないからである。

【００９９】以上のディスク管理テーブル２１０の設定により、後述するｍｏｕｎｔ処理を論理バーチャルアレイディスク装置ｖａ００に対して行うと、ｈｄ０～ｈｄ３の４台の磁気ディスク装置の第０パーティションを論理バーチャルアレイディスク装置ｖａ００として利用することが可能となる。

【０１００】図８はファイル管理テーブル２２０の説明図である。ファイル管理テーブル２２０は各サブファイルごとに割り付けられ、それぞれファイル属性領域と、ディスクブロックインデックス領域に分けられる。各サブファイルのファイル記述子がファイル管理テーブル２２０－０～２２０－ｎの先頭を指し示すことにより、ファイル管理テーブル２２０はファイル記述子と関連付けられている。

【０１０１】ファイル属性領域は、ファイルタイプ、ファイルサイズ、格納デバイス、およびストライピングブロックを格納するエントリから構成される。ファイルタイプには、このテーブルが管理するファイルがバーチャルアレイディスク装置に格納されるサブファイルであるのか、通常の磁気ディスク装置に格納されるファイルであるのかを示す識別子が格納される。ファイルサイズには、サブファイルの容量が格納される。格納デバイスには、サブファイルが格納される磁気ディスク装置の実体である、論理ディスク装置の名称が格納される。ストライピングブロックには、ファイルをこのサブファイルに分割した際に単位としたデータブロックの個数が格納される。すなわち各サブファイルは、このデータブロックの個数ごとにファイルを先頭から分割して作られる。

【０１０２】ディスクブロックインデックス領域には、データブロックのインデックス情報としてディスク上の論理ブロック番号を格納する。本図では、ファイルを構成する各データブロックが論理ブロック番号１００、２００、３００、．．．９００に格納されていることになる。

【０１０３】ファイル管理テーブル２２０はバーチャルアレイディスクに格納するファイルが作られるときに、アクセス制御プログラムによってサブファイルごとに作られる。以後、このファイルが消去されるまでファイル管理テーブルは存続し、ファイルを構成する各サブファイルをアクセスする際にアクセス制御プログラムがこれを参照する。バーチャルアレイディスクに格納されたファイルが消去されるときに各サブファイルが消去されることになり、この時同時にファイル管理テーブルも消去される。

【０１０４】図９はファイル記述子対応テーブル２３０の説明図である。バーチャルアレイディスクに格納されている元ファイルのファイル記述子ｖｆｄと、分割後のサブファイルのファイル記述子ｆｄとの対応を示している。

【０１０５】本図の例では、ｖｆｄ＝４である元ファイルは、４個のサブファイルからなり、それぞれのファイル記述子は、ｆｄ０＝５、ｆｄ１＝６、ｆｄ２＝７、ｆｄ３＝８である。また、ｖｆｄ＝２０の元ファイルについても同様に、サブファイルのファイル記述子はｆｄ０＝２１、ｆｄ１＝２２、ｆｄ２＝２３、ｆｄ３＝２４であることを示している。

【０１０６】バーチャルアレイディスクに格納されているファイルをアクセスする際には、まずファイルをｏｐｅｎしなければならない。この時に、元ファイルのファイル記述子ｖｆｄがアクセス制御プログラムにより割り当てられる。次にその実体であるサブファイルの名称が元ファイルの名称から生成され、このサブファイル名称を用いてサブファイルがｏｐｅｎされる。この時に各サブファイルにファイル記述子ｆｄ０～ｆｄ３が割り当てられる。以後、アプリケーションプログラムからこのファイルをアクセスする場合にはファイル記述子ｖｆｄを用いる。アクセス制御プログラムはファイル記述子対応テーブル２３０を用い、これをサブファイルのファイル記述子ｆｄ０～ｆｄ３に変換して各サブファイルのアクセスを行う。ファイル記述子対応テーブルのエントリは、ファイルをｃｌｏｓｅする際に解放されて内容は無効となる。

【０１０７】以上説明したディスク管理テーブル２１０、ファイル管理テーブル２２０、およびファイル記述子対応テーブル２３０を用いて、バーチャルアレイディスクを制御する際の概略の手順について説明する。

【０１０８】図１０に制御の全体のフローを示す。各ステップは、ユーザからのコマンド、あるいはプログラムからの関数コールによって起動される処理を示している。各処理はメモリに格納されたアクセス制御プログラムによって実行される。以下、本図の制御フローに従ってその内容を説明する。

【０１０９】ｍｏｕｎｔ処理では、バーチャルアレイディスクをＯＳが管理するディレクトリの指定された場所に割り付け、その指定ディレクトリ名称下のデバイスとしてソフトウェアから利用可能な状態とする。すなわち、論理バーチャルアレイディスク装置を構成する論理

ディスク装置を、ディスク管理テーブル２１０を参照してディレクトリの指定された場所に割り付ける処理を行なう。

【０１１０】ｏｐｅｎ処理では、元ファイル及びサブファイルにファイル記述子を割り当て、ファイル記述子対応テーブル２３０に対応関係が取れるように登録する。

【０１１１】ｒｅａｄ／ｗｒｉｔｅ処理では、ｒｅａｄ処理指定時にはバーチャルアレイディスクからのファイル読み出しを行い、ｗｒｉｔｅ処理指定時にはバーチャルアレイディスクへのファイル書き込みを行う。いずれの場合も、元ファイルのファイル記述子からファイル記述子対応テーブル２３０によりサブファイルのファイル記述子を得る。さらにファイル記述子によりファイル管理テーブル２２０を得て、サブファイルの格納されている論理ディスク装置にアクセスする。

【０１１２】ｃｌｏｓｅ処理では、元ファイルとサブファイルのクローズ処理、すなわち元ファイルに割り当てたファイル記述子と、サブファイルに割り当てたファイル記述子の解放を行う。

【０１１３】ｕｍｏｕｎｔ処理では、論理バーチャルアレイディスク装置として指定ディレクトリに割り付けられた各論理ディスク装置を指定ディレクトリから除去する。

【０１１４】このため、その指定ディレクトリ名称下のデバイスとしてソフトウェアから意識できなくなる。

【０１１５】以下、各処理の詳細な制御フローを図１１、図１２、図１３、図１４、図１５、図１６、および図１７を用いて説明する。

【０１１６】ｍｏｕｎｔ処理の制御フロー（１）～（３）を図１１を用いて説明する。なお、／ｄｅｖで始まる装置名称は、ｍｏｕｎｔ処理やｕｍｏｕｎｔ処理で用いられるフルパス名称である。

【０１１７】（１）マウントする論理バーチャルアレイディスク装置に対応する磁気ディスク装置の名称、パーティション番号、すなわち論理ディスク装置の名称をディスク管理テーブル２１０から得る。

【０１１８】（２）論理バーチャルアレイディスク装置をマウントするディレクトリに、マウント用のディレクトリを作る。例えば図１２に示すように論理バーチャルアレイディスク装置の名称が／ｄｅｖ／ｖａ００である場合には、名称が“.ｖａ００”というディレクトリをまずマウントディレクトリ／ｄａｔａの下に作り、さらにその下に、すなわち／ｄａｔａ／.ｖａ００の下に名称が“.ｖａ０００”、“.ｖａ００１”、“.ｖａ００２”、および“.ｖａ００３”というディレクトリを作る。

【０１１９】（３）ディスク管理テーブル２１０を参照して、各ディレクトリ“.ｖａ０００”、“.ｖａ００１”、“.ｖａ００２”、および“.ｖａ００３”に対応する論理ディスク装置／ｄｅｖ／ｈｄ００～／ｄｅｖ／ｈｄ０３をマウントする。

【０１２０】図１２には、ｍｏｕｎｔ処理の結果により構成されるトリー構造が示されている。

【０１２１】このようにｍｏｕｎｔ処理ではトリー構造を持つディレクトリ下にバーチャルアレイディスクがマウントされ、バーチャルアレイディスク内のファイルがトリー構造のディレクトリによって管理される。この例では、“／ｄａｔａ”ディレクトリに論理バーチャルアレイディスク装置“／ｄｅｖ／ｖａ００”がマウントされている。各磁気ディスク装置には同一構造のディレクトリが構成され、その下に各サブファイルが格納されることになる。ここでは、各サブファイル名称とディレクトリ名称に添字を付して区別してある。

【０１２２】ｏｐｅｎ処理の制御フロー（１）～（５）を図１３を用いて説明する。

【０１２３】（１）オープン対象として指定された元ファイルの名称から、添字を付してサブファイルの名称を生成する。

【０１２４】例えば、図１２のディレクトリ構造において、元ファイルとして／ｄａｔａ／ｆｉｌｅを指定すると、これからサブファイルのファイル名称
／ｄａｔａ／.ｖａ００／.ｖａ０００／ｆｉｌｅ０、
／ｄａｔａ／.ｖａ００／.ｖａ００１／ｆｉｌｅ１、
／ｄａｔａ／.ｖａ００／.ｖａ００２／ｆｉｌｅ２、
／ｄａｔａ／.ｖａ００／.ｖａ００３／ｆｉｌｅ３、
を生成する。

【０１２５】（２）サブファイルのファイル名称を用いて個々のサブファイルをオープンし、ファイル記述子ｆｄ０～ｆｄ３を割当てる。これにより、各サブファイルのファイル記述子ｆｄ０～ｆｄ３がファイル管理テーブル２２０に対応付けられ、アプリケーションプログラムからのファイル記述子を用いたアクセスが可能となる。

【０１２６】（３）元ファイルにファイル記述子ｖｆｄを割り当てる。

【０１２７】（４）元ファイルのファイル記述子ｖｆｄとサブファイルのファイル記述子ｆｄ０～ｆｄ３とを対にして、ファイル記述子対応テーブル２３０に登録する。（５）元ファイルのファイル記述子ｖｆｄを、本ｏｐｅｎ処理の結果として、すなわちｏｐｅｎコールの戻り値としてアプリケーションプログラムへ返す。

【０１２８】次にｒｅａｄ／ｗｒｉｔｅ処理でのｒｅａｄ処理実行の制御フロー（１）～（５）について図１４を用いて説明する。

【０１２９】（１）アプリケーションプログラムからは元ファイルのファイル記述子ｖｆｄを引き数にしてアクセス要求が発行される。引数であるファイル記述子ｖｆｄをキーとしてファイル記述子対応テーブル２３０を検索し、ｖｆｄに対応するサブファイルのファイル記述子ｆｄ０～ｆｄ３を得る。図９に示すように例えば元ファイルのファイル記述子ｖｆｄが４の場合、サブファイル

のファイル記述子ｆｄ０～ｆｄ３として５、６、７、８が得られる。

【０１３０】（２）次に、サブファイルのファイル記述子ｆｄ０～ｆｄ３から、これらのサブファイルのファイル管理テーブル２２０を得る。

【０１３１】（３）サブファイルをアクセスする順番は、ディスク管理テーブル２１０の記述子の添字の順に行う。この順にサブファイルのファイル管理テーブル２２０を参照し、読み込むデータブロックの論理ブロック番号を得る。図７に示す例では、論理ディスク装置／ｄｅｖ／ｈｄ００、／ｄｅｖ／ｈｄ０１、／ｄｅｖ／ｈｄ０２、／ｄｅｖ／ｈｄ０３の順にアクセスすることになる。ファイル管理テーブル２２０に書かれたストライピングブロックが指定するデータブロック数だけアクセスしたら、次のサブファイルへアクセスする。すなわち、次のサブファイルのファイル管理テーブルを参照して、ストライピングブロックが指定するデータブロック数だけアクセスする。この手順にしたがって、読み込むデータブロックの論理ブロック番号を順に得る。

【０１３２】（４）磁気ディスク装置から読み込むデータブロックの論理ブロック番号と、サブファイルのファイル管理テーブル２２０の格納デバイスエントリに書かれた論理ディスク装置とにしたがって、データブロックを読み込む。実際にアクセスする磁気ディスク装置は、ディスク管理テーブル２１０を参照して、該当論理ディスク装置をその一部として含む磁気ディスク装置として得ることができる。

【０１３３】（５）最後に、読み込むべきデータブロックをすべて読み込んだかどうか判定する。まだ読み込むべきデータブロックがあれば、（３）に戻って繰り返す。

【０１３４】次にｒｅａｄ／ｗｒｉｔｅ処理でのｗｒｉｔｅ処理実行の制御フロー（１）～（７）について図１５を用いて説明する。サブファイルのデータブロックをアクセスする手順の概略は、ｒｅａｄ処理実行時と同様である。

【０１３５】（１）アプリケーションプログラムからは元ファイルのファイル記述子ｖｆｄを引き数にしてアクセス要求が発行される。引数であるファイル記述子ｖｆｄをキーとしてファイル記述子対応テーブル２３０を検索し、ｖｆｄに対応するサブファイルのファイル記述子ｆｄ０～ｆｄ３を得る。図９に示すように例えば元ファイルのファイル記述子ｖｆｄが４の場合、サブファイルのファイル記述子ｆｄ０～ｆｄ３として５、６、７、８が得られる。

【０１３６】（２）次に、サブファイルのファイル記述子ｆｄ０～ｆｄ３から、これらのサブファイルのファイル管理テーブル２２０を得る。

【０１３７】（３）既存ファイルの内容を更新する場合には、データブロックの内容を上書きすれば良い。しかし、ファイルサイズを超えて追加書き込みを行う場合には、書き込むために新しい領域を割り当てる必要が生じる。したがって、新しいディスクブロックの割当てが必要であるかどうかを判定し、必要であれば（４）へ、不要であれば（５）へ進む。

【０１３８】（４）ディスクブロックを割り当てる際に、どのサブファイルが格納されている論理ディスク装置から割り当てるのかを決める。ファイルの最終部分となっているサブファイルの総データブロック数が、ファイル管理テーブルのストライピングブロックエントリに書かれたデータブロック数の整数倍である場合、次のサブファイルを格納する論理ディスク装置からデータブロックの割り当てを行う。そうでなければ、ファイル最終部分のサブファイルが格納されている論理ディスク装置から割り当てを行う。データブロックを割り当てた後に、そのデータブロックの論理ブロック番号を得る。次は（６）へ進む。

【０１３９】（５）ｒｅａｄ処理の場合で述べたように元ファイルはサブファイルに分割されており、これにより元ファイルに合成するためのアクセスする順番は決定される。したがって、その手順に従い書き込み箇所に該当するサブファイルのファイル管理テーブル２２０から、ファイルが書き込まれるデータブロックの論理ブロック番号を得る。

【０１４０】（６）磁気ディスク装置へ書き込むデータブロックの論理ブロック番号と、サブファイルのファイル管理テーブル２２０の格納デバイスエントリに書かれた論理ディスク装置とにしたがって、データブロックを書き込む。実際にアクセスする磁気ディスク装置は、ディスク管理テーブル２１０を参照して、該当論理ディスク装置をその一部として含む磁気ディスク装置として得ることができる。

【０１４１】（７）最後に、書き込むべきデータブロックをすべて書き込んだかどうか判定する。まだ書き込むべきデータブロックがあれば、（３）に戻って繰り返す。

【０１４２】ｃｌｏｓｅ処理の制御フロー（１）～（３）を図１６を用いて説明する。

【０１４３】（１）アプリケーションプログラムからは元ファイルのファイル記述子ｖｆｄを引き数にして処理要求が発行される。ファイル記述子ｖｆｄをキーとしてファイル記述子対応テーブル２３０を検索し、ｖｆｄに対応するサブファイルのファイル記述子ｆｄ０～ｆｄ３を得る。

【０１４４】（２）サブファイルに対応するファイル記述子ｆｄ０～ｆｄ３を解放する。

【０１４５】（３）元ファイルに対応するファイル記述子ｖｆｄを解放する。

【０１４６】最後に、ｕｍｏｕｎｔ処理の制御フロー（１）～（３）を図１７を用いて説明する。

【０１４７】（１）ディスク管理テーブル２１０を参照して、アンマウントする論理バーチャルアレイディスク装置を構成する論理ディスク装置に対応する、磁気ディスク装置の名称とパーティション番号を得る。

【０１４８】（２）上記、論理ディスク装置をアンマウントする。すなわち図１２の例では、ディレクトリ“.ｖａ０００”、“.ｖａ００１”、“.ｖａ００２”、および“.ｖａ００３”から／ｄｅｖ／ｈｄ００～／ｄｅｖ／ｈｄ３０を除去する。

【０１４９】（３）バーチャルアレイディスクのマウント用に作ったディレクトリを消去する。すなわち図１２の例では、“.ｖａ００”、“.ｖａ０００”、“.ｖａ００１”、“.ｖａ００２”、および“.ｖａ００３”を消去する。

【０１５０】以上説明したように、ディスク管理テーブル２１０によりバーチャルアレイディスクを定義し、ファイル記述子対応テーブル２３０により元ファイルとサブファイルとの対応付けを行い、ファイル管理テーブル２２０を参照してサブファイルへアクセスすることにより、ユーザは複数のサブファイルにアクセスすることを全く意識せず、あたかも単一のファイルにアクセスするのと全く同じ形でバーチャルアレイディスクを利用することができるという効果が得られる。

【０１５１】次に図１８～２１を用いてファイルのストライピング処理について説明する。

【０１５２】図１に示した実施例では、データブロック１個を単位としたファイルのサブファイルへの分割、すなわちストライピングを行っている。この場合、データブロックＡから順にＢ、Ｃ、Ｄを、磁気ディスク装置のｉｄ番号０番から順に１番、２番、３番へと格納する。そして、次のデータブロックＥから順にＦ、Ｇ、Ｈを、同様に磁気ディスク装置のｉｄ番号０番から順に１番、２番、３番へと格納する。この場合のファイルの書き込みの様子を図１８に、ファイルの読み出しの様子を図１９に示す。

【０１５３】図１８はファイルを先頭から順に書き込む場合を示している。Ｉ／Ｏバスインタフェース１５０を介して、ホストＣＰＵ１００からデータブロックが順に送られる。ディスコネクト・リコネクト機能により各磁気ディスク装置がＩ／Ｏバスインタフェース１５０と順に接続されて、データブロック０、１、２、３、４．．．がｉｄ番号０番、１番、２番、３番、０番．．．の磁気ディスク装置へと格納される。なお、格納する磁気ディスク装置の順序は必ずしもｉｄ番号の順である必要はなく、０番以外の磁気ディスク装置から開始することも可能である。また、ｉｄ番号の順序を昇順、または降順以外とすることも可能である。

【０１５４】図１９はファイルを先頭から順に読み出す場合を示している。Ｉ／Ｏバスインタフェース１５０を介して、ホストＣＰＵ１００へデータブロックが順に送られる。図１８に示したようにデータブロックは格納されており、ディスコネクト・リコネクト機能により各磁気ディスク装置がＩ／Ｏバスインタフェース１５０と順に接続されて、データブロック０、１、２、３、４．．．がｉｄ番号０番、１番、２番、３番、０番．．．の磁気ディスク装置から読み出される。なお、格納する磁気ディスク装置の順序は必ずしもｉｄ番号の順である必要はなく、０番以外の磁気ディスク装置から開始することも可能である。また、ｉｄ番号の順序を昇順、または降順以外とすることも可能である。

【０１５５】一方、複数個のデータブロックをストライピングの単位とすることも可能である。一例として、データブロック４個を単位としてストライピングを行った場合のファイルの書き込みの様子を図２０に、ファイルの読み出しの様子を図２１に示す。

【０１５６】図２０はデータブロック４個を単位としてファイルを先頭から順に書き込む場合を示している。Ｉ／Ｏバスインタフェース１５０を介して、ホストＣＰＵ１００からデータブロックが順に送られる。ディスコネクト・リコネクト機能により各磁気ディスク装置がＩ／Ｏバスインタフェース１５０と順に接続されて、データブロック０、１、２、３がｉｄ番号０番、データブロック４、５、６、７がｉｄ番号１番、データブロック８、９、１０、１１がｉｄ番号２番、データブロック１２、１３、１４、１５がｉｄ番号３番、データブロック１６、１７、１８、１９がｉｄ番号０番．．．の磁気ディスク装置へとそれぞれ格納される。なお、格納する磁気ディスク装置の順序は必ずしもｉｄ番号の順である必要はなく、０番以外の磁気ディスク装置から開始することも可能である。また、ｉｄ番号の順序を昇順、または降順以外とすることも可能である。

【０１５７】図２１はデータブロック４個を単位としてファイルを先頭から順に読み出す場合を示している。Ｉ／Ｏバスインタフェース１５０を介して、ホストＣＰＵ１００へデータブロックが順に送られる。図２０に示したようにデータブロックは格納されており、ディスコネクト・リコネクト機能により各磁気ディスク装置がＩ／Ｏバスインタフェース１５０と順に接続されて、データブロック０、１、２、３がｉｄ番号０番、データブロック４、５、６、７がｉｄ番号１番、データブロック８、９、１０、１１がｉｄ番号２番、データブロック１２、１３、１４、１５がｉｄ番号３番、データブロック１６、１７、１８、１９がｉｄ番号０番．．．の磁気ディスク装置からとそれぞれ読み出される。なお、格納する磁気ディスク装置の順序は必ずしもｉｄ番号の順である必要はなく、０番以外の磁気ディスク装置から開始することも可能である。また、ｉｄ番号の順序を昇順、または降順以外とすることも可能である。

【０１５８】なお、ストライピングの単位とするデータブロック数は、４個以外の任意の個数を取りうることは

言うまでもない。

【０１５９】図２２～３３を用いて、論理バーチャルアレイディスク装置として使用する論理ディスク装置の組合せについて説明する。

【０１６０】本実施例では、ディスク管理テーブル２１０で磁気ディスク装置のｉｄ番号または名称と、パーティション番号とを指定して使用する論理ディスク装置を定義する。図７の説明で述べたように、複数の論理ディスク装置に一連の識別子ｖａＸＹ０、ｖａＸＹ１、．．．を付けることで、論理バーチャルアレイディスク装置として利用する論理ディスク装置の組合せの定義が可能である。識別子の添字の意味は、図７のディスク管理テーブルの説明で述べたとおりである。本実施例ではサブファイルの物理的な配置の自由度が非常に高く、以下のような構成が可能である。なお、例えば図７のディスク装置名称ｈｄ４、ｈｄ５のディスク装置のように、論理バーチャルアレイディスク装置として定義していない領域は、特に断わりがなくとも通常の論理ディスク装置として利用可能である。

【０１６１】図２２はｉｄ番号０、１、２、３の４台の磁気ディスク装置のすべてのパーティションを論理バーチャルアレイディスク装置として用いる場合である。

【０１６２】これに対応するディスク管理テーブル２１０の設定例を図２３に示す。４台の磁気ディスク装置をバーチャルアレイディスク装置ｖａ０として定義し、全体をひとつの論理バーチャルアレイディスク装置ｖａ００として用いる場合に相当する。ｉｄ番号４、５、６の３台の磁気ディスク装置は、通常の磁気ディスク装置として用いる。

【０１６３】図２４はｉｄ番号０、１、２、３の４台の磁気ディスク装置の特定のパーティションを用いる場合である。

【０１６４】これに対応するディスク管理テーブルの設定例を図２５に示す。図２３と同様に定義したバーチャルアレイディスク装置ｖａ０の第０パーティションを論理バーチャルアレイディスク装置ｖａ００として用いる場合である。ｉｄ番号４、５、６の３台の磁気ディスク装置は、通常の磁気ディスク装置として用いる。

【０１６５】図２６はｉｄ番号０、１、２、３の４台の磁気ディスク装置の各３つのパーティションを用いて、論理バーチャルアレイディスク装置を３組定義した場合である。

【０１６６】これに対応するディスク管理テーブル２１０の設定例を図２７に示す。図２３と同様に定義したバーチャルアレイディスク装置ｖａ０の第０、第１、第２パーティションを、それぞれ論理バーチャルアレイディスク装置ｖａ００、ｖａ０１、ｖａ０２として用いる場合である。ｉｄ番号４、５、６の３台の磁気ディスク装置は、通常の磁気ディスク装置として用いる。

【０１６７】図２８はｉｄ番号０、１、２、３の４台の磁気ディスク装置の特定のパーティション、およびｉｄ番号４、５の２台の磁気ディスク装置の特定のパーティションを用い、論理バーチャルアレイディスク装置を２組定義した場合である。

【０１６８】これに対応するディスク管理テーブル２１０の設定例を図２９に示す。ｉｄ番号０、１、２、３の４台の磁気ディスク装置をバーチャルアレイディスク装置ｖａ０、ｉｄ番号４、５の２台の磁気ディスク装置をバーチャルアレイディスク装置ｖａ１としてそれぞれ定義し、各々の第０パーティションをそれぞれ論理バーチャルアレイディスク装置ｖａ００、ｖａ１０として用いる場合である。ｉｄ番号６の磁気ディスク装置は、通常の磁気ディスク装置として用いる。

【０１６９】図３０はｉｄ番号０、１、２、３の４台の磁気ディスク装置の各３つのパーティション、およびｉｄ番号４、５の２台の磁気ディスク装置の各３つのパーティションを用い、論理バーチャルアレイディスク装置を６組定義した場合である。

【０１７０】これに対応するディスク管理テーブル２１０の設定例を図３１に示す。図２９と同様に定義したバーチャルアレイディスク装置ｖａ０、ｖａ１の第０、第１、第２パーティションを、それぞれ論理バーチャルアレイディスク装置ｖａ００、ｖａ０１、ｖａ０２、およびｖａ１０、ｖａ１１、ｖａ１２として用いる場合である。ｉｄ番号６の磁気ディスク装置は通常の磁気ディスク装置として用いる。

【０１７１】図３２はｉｄ番号０、２の２台の磁気ディスク装置の各３つのパーティション、ｉｄ番号４、６の２台の磁気ディスク装置の各２つのパーティション、およびｉｄ番号１、３の２台の磁気ディスク装置の各１つのパーティションを用い、論理バーチャルアレイディスク装置を３組定義した場合である。

【０１７２】これに対応するディスク管理テーブル２１０の設定例を図３３に示す。ｉｄ番号０、２、４、６の４台の磁気ディスク装置をバーチャルアレイディスク装置ｖａ０、ｉｄ番号０、１、２、３の４台の磁気ディスク装置をバーチャルアレイディスク装置ｖａ１としてそれぞれ定義し、ｖａ０の第０、第２パーティションをそれぞれ論理バーチャルアレイディスク装置ｖａ００、ｖａ０２として、ｖａ１の第１パーティションを論理バーチャルアレイディスク装置ｖａ１１として用いる場合である。ｉｄ番号０、２の２台の磁気ディスク装置はｖａ０とｖａ１とに重複して定義されることになるが、利用するパーティションが重ならないようにすれば問題は生じない。なお、ｉｄ番号５の磁気ディスク装置は、通常の磁気ディスク装置として用いる。　以上の各構成例において、磁気ディスク装置の台数及びｉｄ番号は、上記以外の組合せでも実現可能である。

【０１７３】図３４～３９は本実施例においてミラーモードを実現する例である。すなわち、論理バーチャルア

レイディスク装置としてプライマリ（正）とバックアップ（副）を一組にして定義し、同一ファイルを両方に格納するものである。これにより、プライマリに障害が発生した場合には、バックアップを替わりに使用することにより信頼性向上を図ることができる。

【０１７４】ファイル書き込みの場合には、プライマリとバックアップの両方に書き込む。

【０１７５】まずプライマリとして定義した論理バーチャルアレイディスク装置に対して、非同期書き込みによりデータブロックを書き込む。引き続き、バックアップとして定義した論理バーチャルアレイディスク装置に対しても、やはり非同期書き込みによりプライマリと同様のデータブロックを書き込む。バックアップへの書き込み命令を発行した後、次のデータブロックの書き込み処理に移り、プライマリおよびバックアップに対して上記と同様の手順によりデータブロックを書き込む。

【０１７６】なお、プライマリとバックアップへの書き込み方に関しては、各論理バーチャルアレイディスク装置を構成する対応するディスク装置ごとに、すなわちｈｄ０とｈｄ２、ｈｄ１とｈｄ３というように書き込むことも、その逆にｈｄ０とｈｄ３、ｈｄ１とｈｄ２というように書き込むことも可能である。これは、ディスク管理テーブルに設定された識別子ｖａＸＹＺのＺの値をキーとして対応を取ることにより行う。具体的な例は図３５の説明で述べる。

【０１７７】これにより、ユーザにはバックアップへの書き込み時間をほとんど感じさせずに、ミラーモードを実現することができる。

【０１７８】ファイル読み出しの場合には、プライマリのみから読み出しを行う。すなわち、ミラーモード指定を行っていない論理バーチャルアレイディスク装置に格納されたファイルを読み出すのと同様の方法で、プライマリからファイルの読み出しを行う。しかし、プライマリに障害が発生すると、バックアップに格納されたファイルを替わりに使用する。

【０１７９】これにより、磁気ディスク装置の信頼性を向上することができる。また、通常の利用状態ではプライマリのみからファイルを読み出すので、バックアップが存在することによるペナルティが発生することなく、高速にファイルの読み出しが行える。

【０１８０】以下、図を用いて、ミラーモードでの論理バーチャルアレイディスク装置として使用する論理ディスク装置の組合せについて説明する。ミラーモードでない論理バーチャルアレイディスク装置の定義の場合と同様に、ディスク管理テーブル２１０により使用する論理ディスク装置に一連の識別子を付けることで定義を行う。さらに、プライマリであるのかバックアップであるのかということと、その組合せを拡張子により示す。例えば、ｖａＸＹＺ＿ｐ０は識別子ｖａＸＹＺの論理ディスク装置がシステム中でミラーモードのプライマリ論理バーチャルアレイディスク装置の構成要素として定義されており、バックアップ論理バーチャルアレイディスク装置の構成要素として定義されるｖａＸＹＺ＿ｂ０の論理ディスク装置と対をなすということを示す。なお、論理バーチャルアレイディスク装置として定義していない領域は、特に断わりがなくとも通常の論理ディスク装置として利用可能である。

【０１８１】図３４はｉｄ番号０、１の２台の磁気ディスク装置の各１つのパーティションを用いてプライマリとし、ｉｄ番号２、３の２台の磁気ディスク装置の各１つのパーティションを用いてバックアップとして、ミラーモードの論理バーチャルアレイディスク装置を１組定義した場合である。

【０１８２】これに対応するディスク管理テーブル２１０の設定例を図３５に示す。ｉｄ番号０、１の２台の磁気ディスク装置をバーチャルアレイディスク装置ｖａ０、ｉｄ番号２、３の２台の磁気ディスク装置をバーチャルアレイディスク装置ｖａ１としてそれぞれ定義し、各々の第０パーティションをそれぞれ論理バーチャルアレイディスク装置ｖａ００＿ｐ０（プライマリ）、ｖａ１０＿ｂ０（バックアップ）として用いる場合である。本図の例では、プライマリを構成する論理ディスク装置ｖａ０００－ｐ０、ｖａ００１－ｐ０に、バックアップを構成する論理ディスク装置ｖａ１００－ｂ０、ｖａ１０１－ｂ０がそれぞれ対応する。ｉｄ番号４、５、６の磁気ディスク装置は、通常の磁気ディスク装置として用いる。

【０１８３】図３６はｉｄ番号０、１、２、３の４台の磁気ディスク装置の各２つのパーティションを用いて、第１パーティションをプライマリとし、第２パーティションをバックアップとして、ミラーモードの論理バーチャルアレイディスク装置を１組定義した場合である。

【０１８４】これに対応するディスク管理テーブル２１０の設定例を図３７に示す。ｉｄ番号０、１、２、３の４台の磁気ディスク装置をバーチャルアレイディスク装置ｖａ１として定義し、第１、第２パーティションをそれぞれ論理バーチャルアレイディスク装置ｖａ１１＿ｐ１（プライマリ）、ｖａ１２＿ｂ１（バックアップ）として用いる場合である。ｉｄ番号４、５、６の磁気ディスク装置は、通常の磁気ディスク装置として用いる。

【０１８５】図３８はｉｄ番号０、１、２、３の４台の磁気ディスク装置のひとつのパーティションを用いてプライマリとし、ｉｄ番号３、４、５、６の４台の磁気ディスク装置のひとつのパーティションを用いてバックアップとして、ミラーモードの論理バーチャルアレイディスク装置を１組定義した場合である。

【０１８６】これに対応するディスク管理テーブル２１０の設定例を図３９に示す。ｉｄ番号０、１、２、３の４台の磁気ディスク装置をバーチャルアレイディスク装置ｖａ２、ｉｄ番号３、４、５、６の４台の磁気ディス

ク装置をバーチャルアレイディスク装置ｖａ３としてそれぞれ定義し、各々の第３、第４パーティションをそれぞれ論理バーチャルアレイディスク装置ｖａ２３＿ｐ２（プライマリ）、ｖａ３４＿ｂ２（バックアップ）として用いる場合である。

【０１８７】図４０は本発明の第３の実施例である。図１に示した第１の実施例におけるｉｄ番号０、１、２、３の磁気ディスク装置１７０－０～１７０－３をアレイディスク装置３００－０～３００－３に置き換えたものである。アレイディスク装置は磁気ディスク装置に比べて装置単体でのファイル転送性能が高いので、本実施例は第１の実施例に比べてより一層ファイルアクセス高速化の効果が得られる。

【０１８８】なお、動作や制御方式は図１の第１の実施例と全く同様である。

【０１８９】これに対応する、アレイディスク装置３００－０～３００－３を含むバーチャルアレイディスクのディスク管理テーブルの設定例を図４１に示す。ｉｄ番号０、１、２、３の４台がアレイディスク装置となっている。ここでは４台のアレイディスク装置ａｄ０、ａｄ１、ａｄ２、およびａｄ３でバーチャルアレイディスク装置ｖａ０を構成している。

【０１９０】また本実施例においても、論理バーチャルアレイディスク装置ｖａ０１＿ｐ０とｖａ０２＿ｂ０、ｖａ２１＿ｐ１とｖａ２２＿ｂ１のようにミラーモードに対応することも可能である。

【０１９１】これまでの実施例では７台の磁気ディスク装置が接続された場合を例に取り説明したが、磁気ディスク装置の台数を２台、３台、４台．．．と順次増やして行くことも可能である。ここで、例えば４台の磁気ディスク装置を５台に増やした場合、４台のディスクに格納されていたファイルを５台の磁気ディスク装置に再配置することが必要となる。

【０１９２】本発明によれば、磁気ディスク装置の増設後に新たなパーティションへファイルをコピーすることにより、ファイルのストライピング数を自動的に増加させて行くことも可能となる。すなわち、バーチャルアレイディスク装置に格納されたファイルを読み書きする場合、ユーザはファイルの物理的な分割を意識することなく処理を行うことができるため、コピー元からコピー先へファイルをコピーするだけで、ユーザがこれを特に意識することなく、ストライピング数の変換を行うことが可能である。この場合について以下に説明する。

【０１９３】図４２にディスク管理テーブル２１０の設定例を、図４３にディスク増設時のファイルコピーの制御フローをそれぞれ示す。図４２において、磁気ディスク装置のｉｄ番号０、１、２、３の第０パーティションにより構成されている論理バーチャルアレイディスク装置ｖａ００はコピー元であり、磁気ディスク装置のｉｄ番号０、１、２、３、４の第１パーティションにより構成されている論理バーチャルアレイディスク装置ｖａ１１はコピー先である。

【０１９４】図４３に示す制御フローに従って、ｖａ００からｖａ１１へファイルを固定長のデータに分割してコピーする。

【０１９５】（１）コピー先の論理バーチャルアレイディスク装置ｖａ１１をディスク管理テーブル２１０に定義する。

【０１９６】（２）コピー元の論理バーチャルアレイディスク装置ｖａ００から、図１４のファイル読み出しの制御フローによりファイルを分割した固定長のデータを読み込み、これをコピー先の論理バーチャルアレイディスク装置ｖａ１１へ、図１５のファイル書き込みの制御フローにより書き込む。

【０１９７】（３）コピー終了か判定し、まだ残りがあれば（２）に戻って繰り返す。コピー終了であれば（４）へ進む。

【０１９８】（４）ディスク管理テーブル２１０上のコピー元の論理バーチャルアレイディスク装置ｖａ００を無効化する。

【０１９９】以上により、磁気ディスク装置の増設に伴うストライピング数の変更が容易に可能となる。この際に、読み出しおよび書き込みの単位とする固定長のデータとしては、任意のブロック数を定義することができる。

【０２００】図４４は複数のＩ／Ｏバスを有する本発明の一実施例である。図１の第１の実施例の構成ではＩ／Ｏバスは１本であったが、図４４では４本のＩ／Ｏバスを有する構成となっている。したがって、Ｉ／Ｏバスインタフェース１５０、１５１、１５２、１５３とＩ／Ｏバス１６０、１６１、１６２、１６３、さらに各Ｉ／Ｏバスごとの７台の磁気ディスク装置１７０－０～１７０－６、１７１－０～１７１－６、１７２－０～１７２－６、１７３－０～１７３－６からなる４組のディスクサブシステムを設けている。アクセス制御プログラムが複数のＩ／Ｏバスへのスケジューリングを行う点を除けば、各ディスクサブシステムの制御方法は前述の各実施例のものをそのまま適用することが可能である。例えば、各Ｉ／Ｏバスごとに第１の実施例のような多重アクセス制御を行い、さらにＩ／Ｏバス間での１階層上での多重アクセス制御を行うということが可能である。また、各Ｉ／Ｏバスごとの多重アクセス制御を並列に行うことも可能である。

【０２０１】各Ｉ／Ｏバス１６０、１６１、１６２、１６３と共通データバス１０１間でのデータ転送において、共通データバスの速度がＩ／Ｏバスの速度の４倍よりも大きく、なおかつ上述のような制御方式を用いることによりデータ転送の相手が特定のＩ／Ｏバスに集中することがないような場合には、本図の実施例は最大のデータ転送速度を達成することができる。その場合、Ｉ／

Ｏバスが１本の第１の実施例に比較して本図の実施例では４倍のデータ転送速度となる。したがって、本実施例によりバーチャルアレイディスクの高速性をさらに高めることが可能となる。なお、本図ではＩ／Ｏバスが４本の場合について示したが、もちろん４本以外の構成を取ることも可能である。

【０２０２】また、磁気ディスク装置を増設してシステムを拡張する場合には、各Ｉ／Ｏバス内での磁気ディスク装置の増設と、Ｉ／Ｏバス単位での増設が可能である。このため、１スタックごとに磁気ディスク装置の増設を行わなければならないアレイディスク装置に比べ、よりフレキシブルにシステムの拡張に対応することができるという長所を有する。

【０２０３】この様に本実施例によれば、バーチャルアレイディスク装置を用いたファイルアクセスの一層の高速化と、柔軟なシステムの拡張性を同時に実現することができるという効果がある。

【０２０４】図４０の第３の実施例に示したように、図４４の実施例の磁気ディスク装置をアレイディスク装置に置き換えた構成を取ることも可能である。この実施例を図４５に示す。図４５では、第０番のＩ／Ｏバスの磁気ディスク装置がアレイディスク装置３００－０～３００－６となっている。これにより、より一層のファイルアクセス高速化を図ることが可能となる。この際、原理的にはアレイディスク装置は任意の磁気ディスク装置と入れ替えることが可能であり、本図に示した以外の構成も容易に実現することが可能である。

【０２０５】なお、上記図４４、および図４５に示した複数のＩ／Ｏバスを有する実施例においても、１本のＩ／Ｏバスを有する実施例の説明中で述べた本発明の種々の機能を実現することが可能なことは明らかである。

【０２０６】本発明によれば、ディスコネクト・リコネクト機能を備えたＩ／Ｏバスを有する計算機システムにおいて、該Ｉ／Ｏバスに接続された複数のディスク装置にファイルを分割格納してソフトウェアで多重・並列アクセス制御することで、高価なハードウェアを用いることなく、アレイディスク装置に匹敵するほどの高速なファイルアクセスを実現することができるという効果がある。しかも、ユーザはあたかも１台のディスク装置を使っているかのごとく、複数のディスクに分割されたファイルをアクセスすることができるようになる。これにより、従来よりも非常に低コストで高速ファイルアクセス可能な計算機システムを実現することができるという効果が得られる。

【０２０７】また、特殊なハードウェアを用いないので、ディスク装置の増設の際には１台づつ増設していくことができ、柔軟なシステム拡張が可能であるという効果もある。

【０２０８】

【発明の効果】本発明によれば、高速なデータアクセス可能な計算機システムを実現することができるという効果が得られる。

【図面の簡単な説明】

【図１】第１の実施例を示す図

【図２】第１の実施例におけるファイル読み出しのタイムチャート

【図３】第１の実施例におけるファイル読み出しのフローチャート

【図４】第１の実施例におけるファイル書き込みのタイムチャート

【図５】第１の実施例におけるファイル書き込みのフローチャート

【図６】第２の実施例を示す図

【図７】ディスク管理テーブルを示す図

【図８】ファイル管理テーブルを示す図

【図９】ファイル記述子対応テーブルを示す図

【図１０】ファイルアクセスの全体フローチャート

【図１１】ｍｏｕｎｔ処理の制御フローチャート

【図１２】ディレクトリ構造を示す図

【図１３】ｏｐｅｎ処理の制御フローチャート

【図１４】ｒｅａｄ処理の制御フローチャート

【図１５】ｗｒｉｔｅ処理の制御フローチャート

【図１６】ｃｌｏｓｅ処理の制御フローチャート

【図１７】ｕｍｏｕｎｔ処理の制御フローチャート

【図１８】データブロック１個を単位とするストライピングを示す図

【図１９】データブロック１個を単位とするストライピングを示す図

【図２０】データブロック４個を単位とするストライピングを示す図

【図２１】データブロック４個を単位とするストライピングを示す図

【図２２】バーチャルアレイディスクの使用領域を示す図

【図２３】ディスク管理テーブルの設定例を示す図

【図２４】バーチャルアレイディスクの使用領域を示す図

【図２５】ディスク管理テーブルの設定例を示す図

【図２６】バーチャルアレイディスクの使用領域を示す図

【図２７】ディスク管理テーブルの設定例を示す図

【図２８】バーチャルアレイディスクの使用領域を示す図

【図２９】ディスク管理テーブルの設定例を示す図

【図３０】バーチャルアレイディスクの使用領域を示す図

【図３１】ディスク管理テーブルの設定例を示す図

【図３２】バーチャルアレイディスクの使用領域を示す図

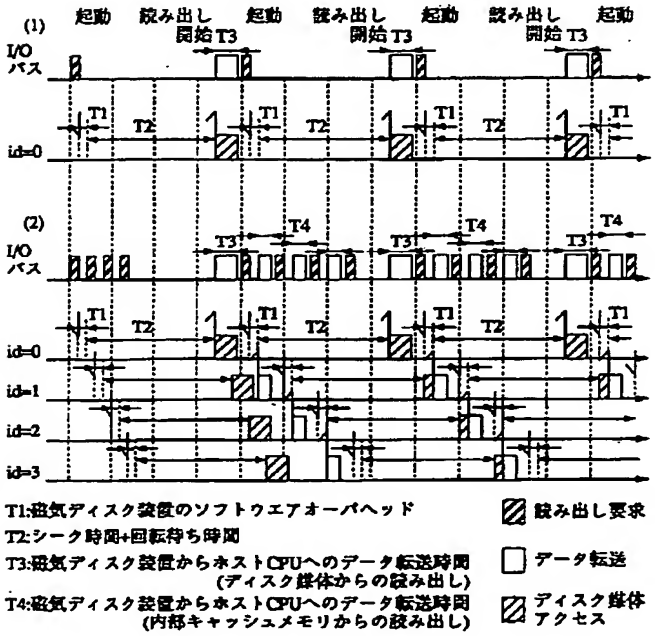
【図３３】ディスク管理テーブルの設定例を示す図

【図３４】ミラーモードの使用領域を示す図
【図３５】ミラーモードでのディスク管理テーブルの設定例を示す図
【図３６】ミラーモードの使用領域を示す図
【図３７】ミラーモードでのディスク管理テーブルの設定例を示す図
【図３８】ミラーモードの使用領域を示す図
【図３９】ミラーモードでのディスク管理テーブルの設定例を示す図
【図４０】アレイディスクを含む構成の第３の実施例を示す図
【図４１】アレイディスクを含むディスク管理テーブルの設定例を示す図
【図４２】ディスク増設時のファイルコピーのディスク管理テーブルの設定例を示す図
【図４３】ディスク増設時のファイルコピーの制御フローチャート
【図４４】複数のＩ／Ｏバスを有する実施例を示す図
【図４５】複数のＩ／Ｏバスを有しアレイディスクを含む実施例を示す図
【図４６】アレイディスクの構成を示す図
【図４７】アレイディスクのタイムチャート
【図４８】本発明の原理図
【図４９】本発明におけるファイル読み出しのタイムチャート

【符号の説明】
１０…ファイル、１０－０～１０－９…ファイルを構成するデータブロック、２０－０～２０－３…磁気ディスク装置の内部キャッシュメモリ、１００…ＣＰＵ、１０１…共通データバス、１１０…メモリ、１２０…ＣＲＴディスプレイ、１３０…キーボード、１４０…ネットワークインタフェース、１５０、１５１、１５２、１５３…Ｉ／Ｏバスインタフェース、１６０、１６１、１６２、１６３…Ｉ／Ｏバス、１７０－０～１７０－６…磁気ディスク装置、１７１－０～１７１－６…磁気ディスク装置、１７２－０～１７２－６…磁気ディスク装置、１７３－０～１７３－６…磁気ディスク装置、１８０－０～１８０－７…ディスコネクト／リコネクト手段、１８１－７、１８２－７、１８３－７…ディスコネクト／リコネクト手段、１９０…ワークステーション、１９１…ＬＡＮ、２００…アクセス制御プログラム、２１０…ディスク管理テーブル、２２０…ファイル管理テーブル、２３０…ファイル記述子対応テーブル、３００、３００－０～３００－６…アレイディスク装置、３１０…ＦＩＦＯ０、３１１…ＦＩＦＯ１、３１２…ＦＩＦＯ２、３１３…ＦＩＦＯ３、３２０…内部バス、３３０…バッファ、３４０…ＳＣＳＩバスインタフェース、４００…ホストＣＰＵ装置。

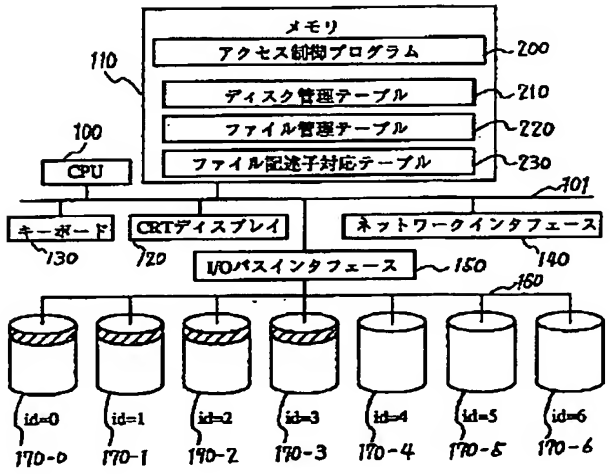
【図２】

図２



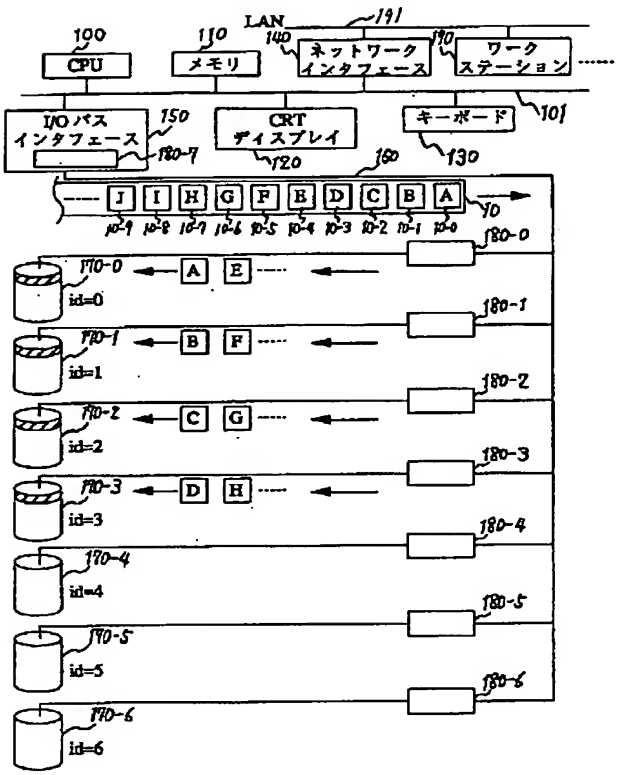
【図６】

図６



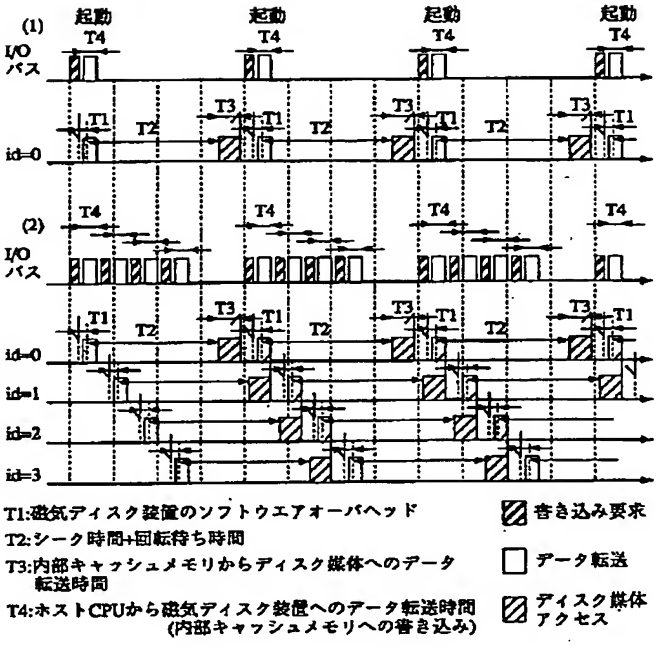
【図１】

図1



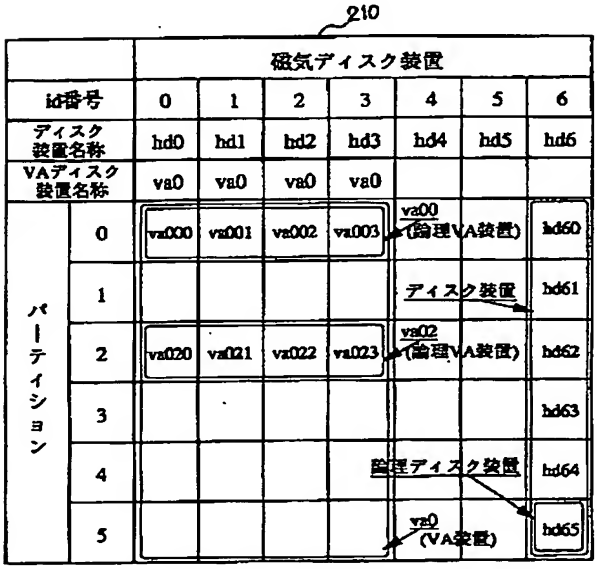
【図４】

図4



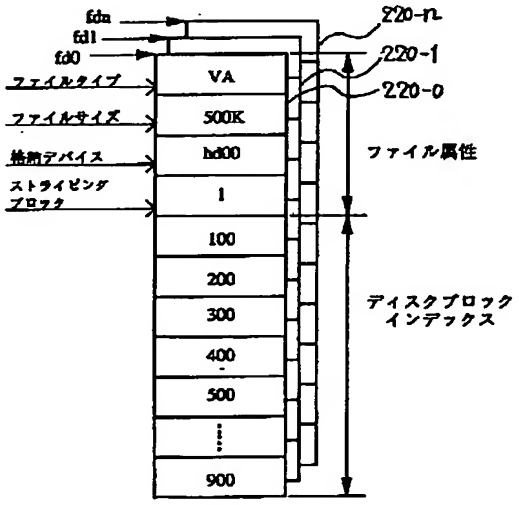
【図７】

図7



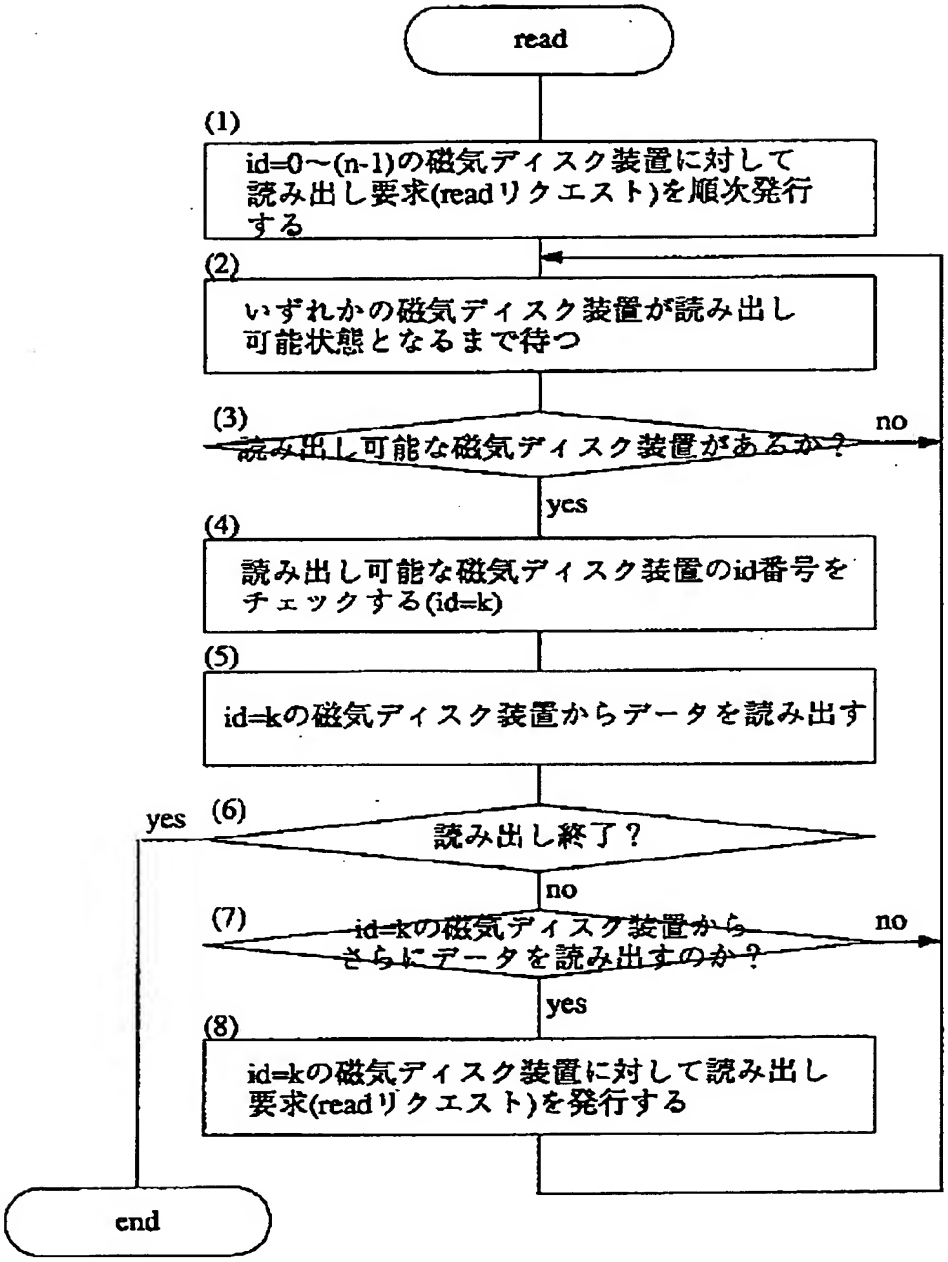
【図８】

図8



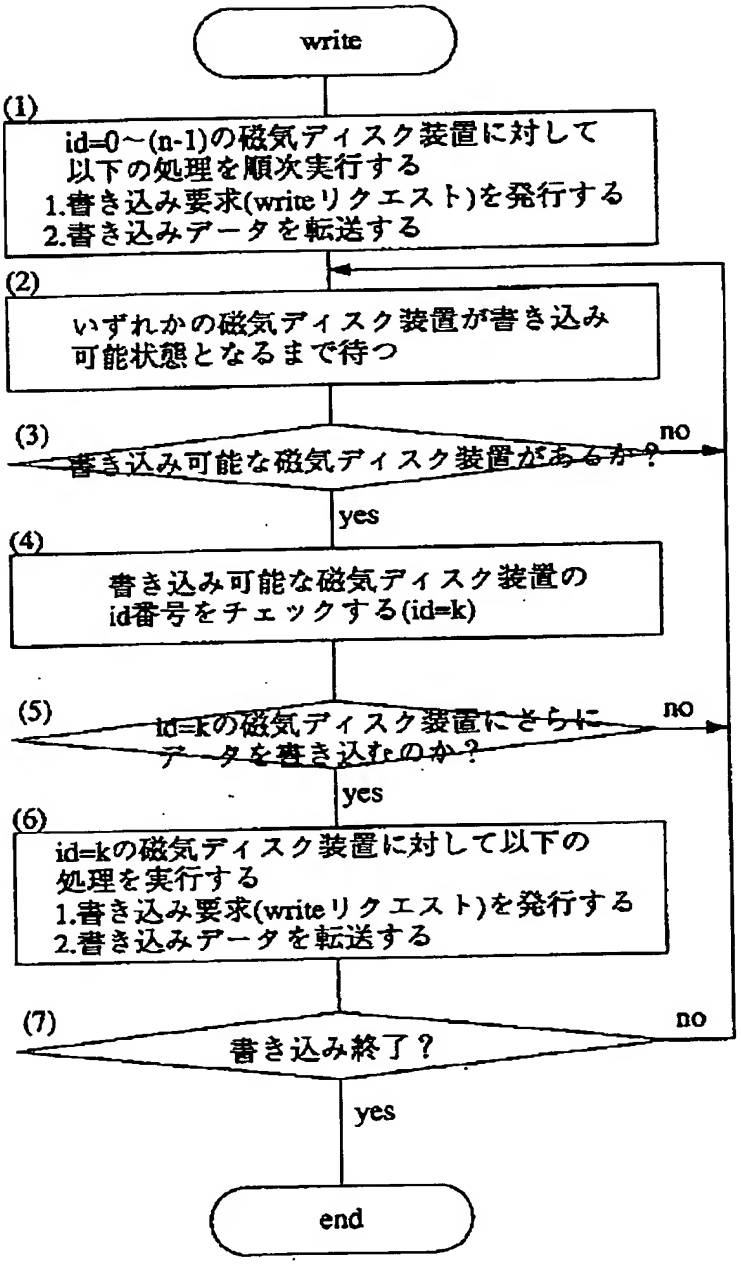
【図３】

図３


read
(1)
id=0～(n-1)の磁気ディスク装置に対して読み出し要求(readリクエスト)を順次発行する
(2)
いずれかの磁気ディスク装置が読み出し可能状態となるまで待つ
(3)
読み出し可能な磁気ディスク装置があるか？
no
yes
(4)
読み出し可能な磁気ディスク装置のid番号をチェックする(id=k)
(5)
id=kの磁気ディスク装置からデータを読み出す
yes
(6)
読み出し終了？
no
(7)
id=kの磁気ディスク装置からさらにデータを読み出すのか？
no
yes
(8)
id=kの磁気ディスク装置に対して読み出し要求(readリクエスト)を発行する
end

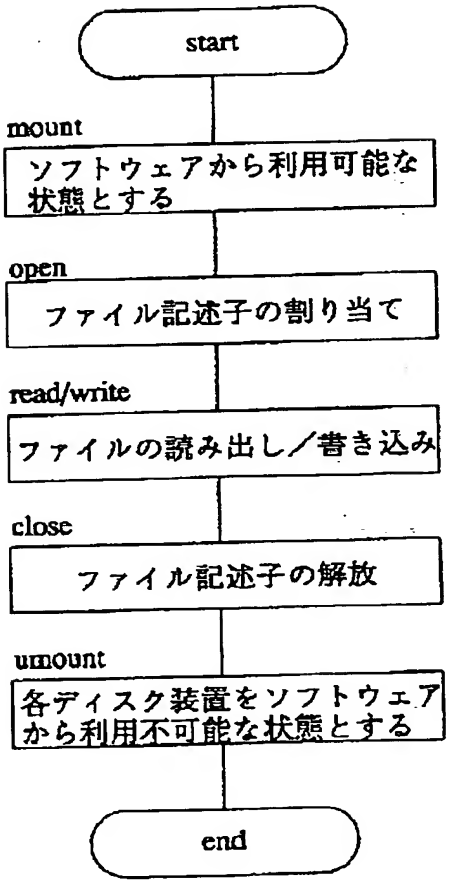
【図５】

図５



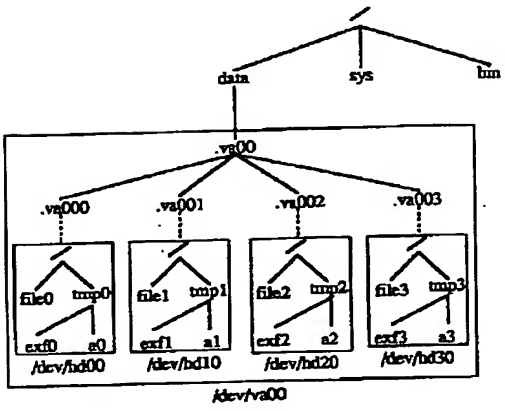
【図１０】

図10



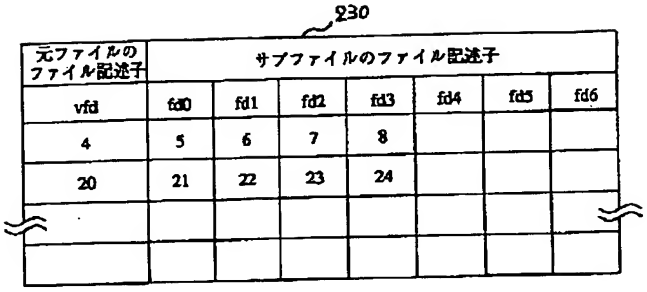
【図１２】

図12



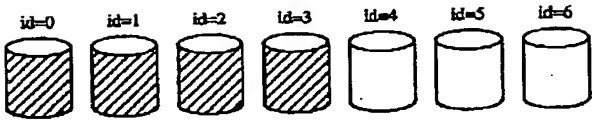
【図９】

図9



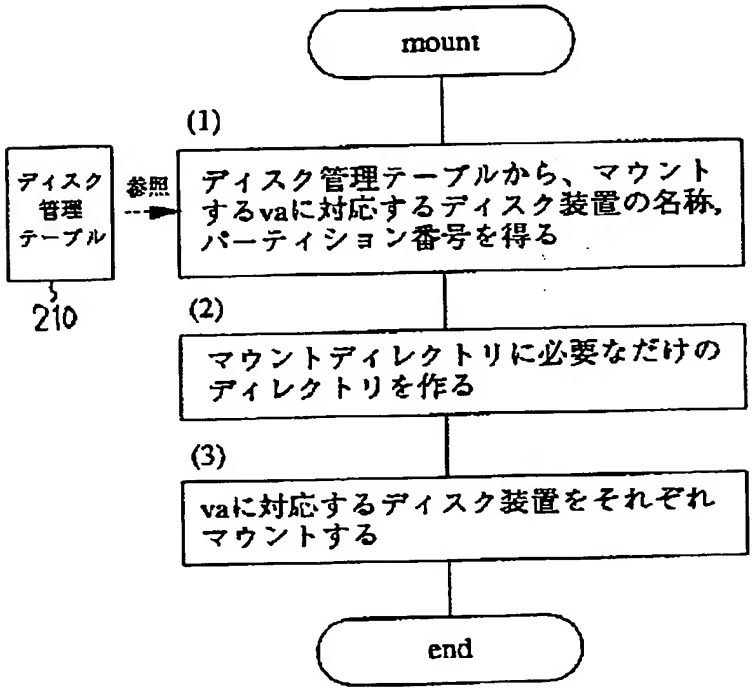
【図２２】

図22



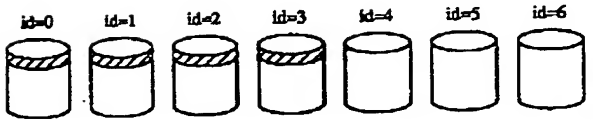
【図１１】

図11



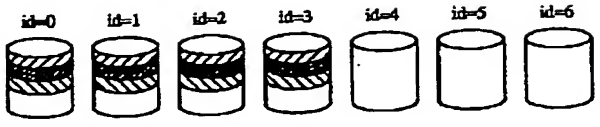
【図２４】

図24



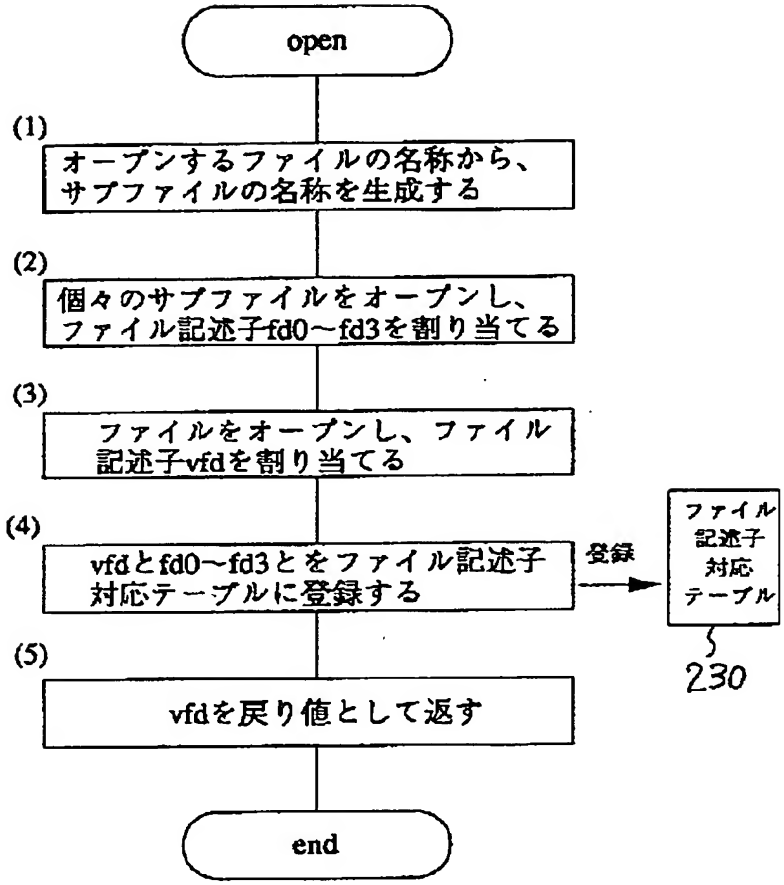
【図２６】

図26



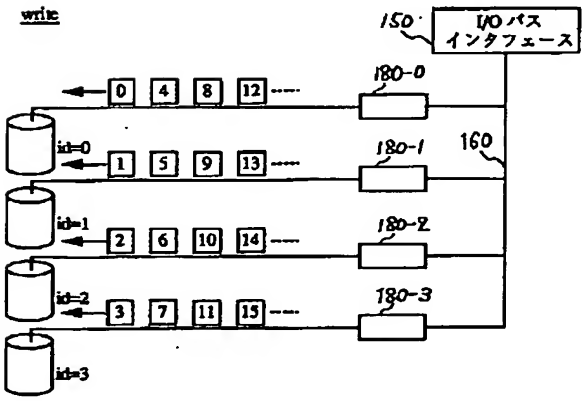
【図１３】

図13



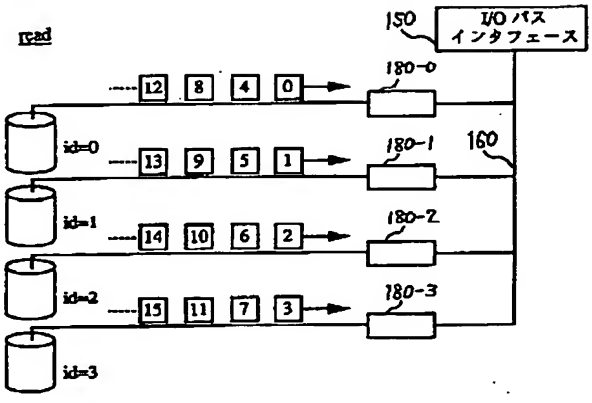
【図１８】

図18



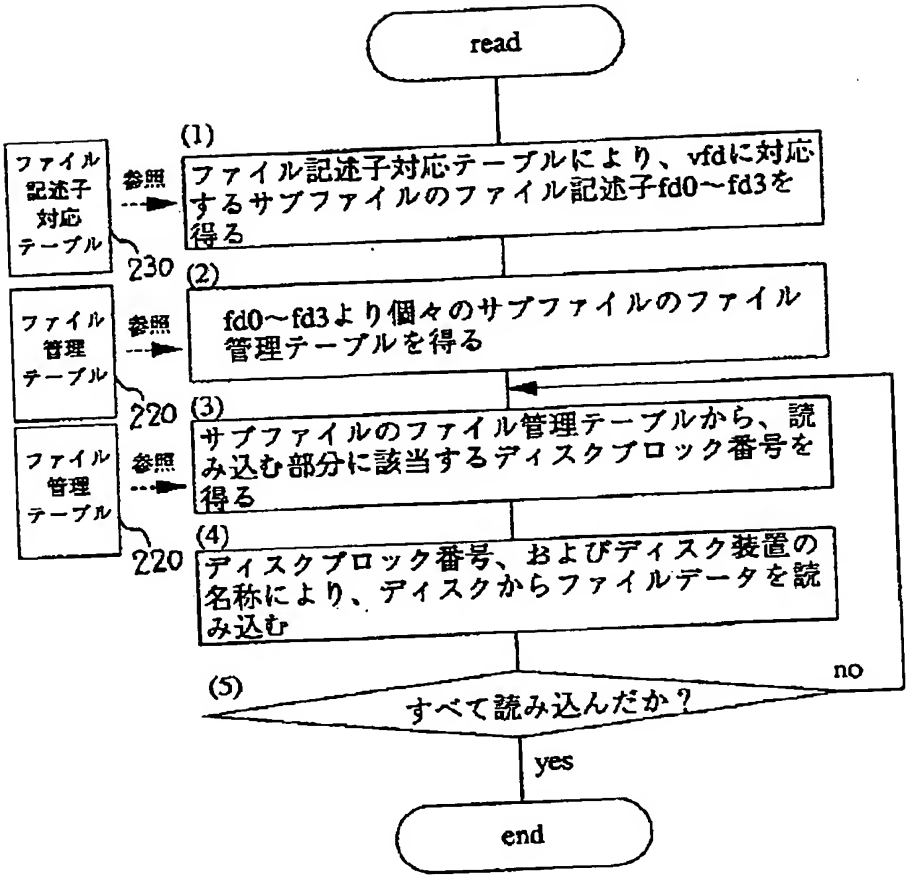
【図１９】

図19



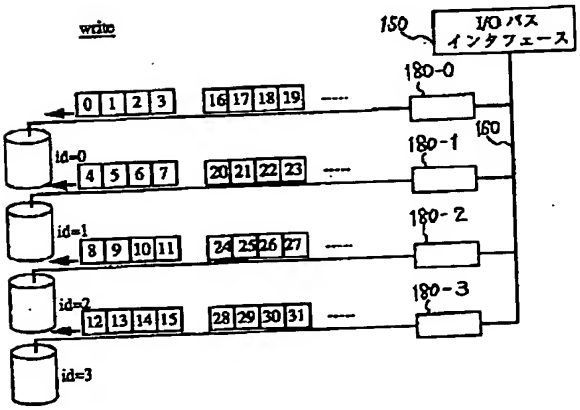
【図１４】

図14



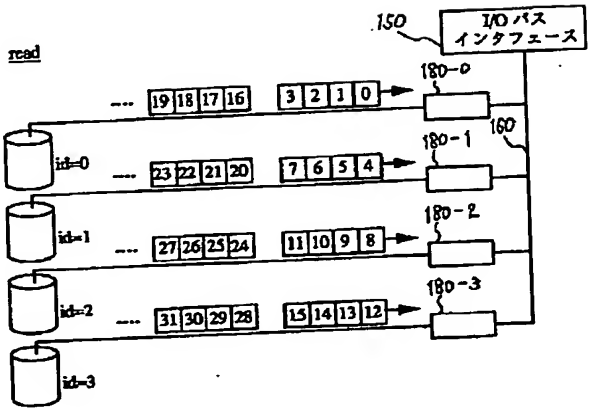
【図２０】

図20



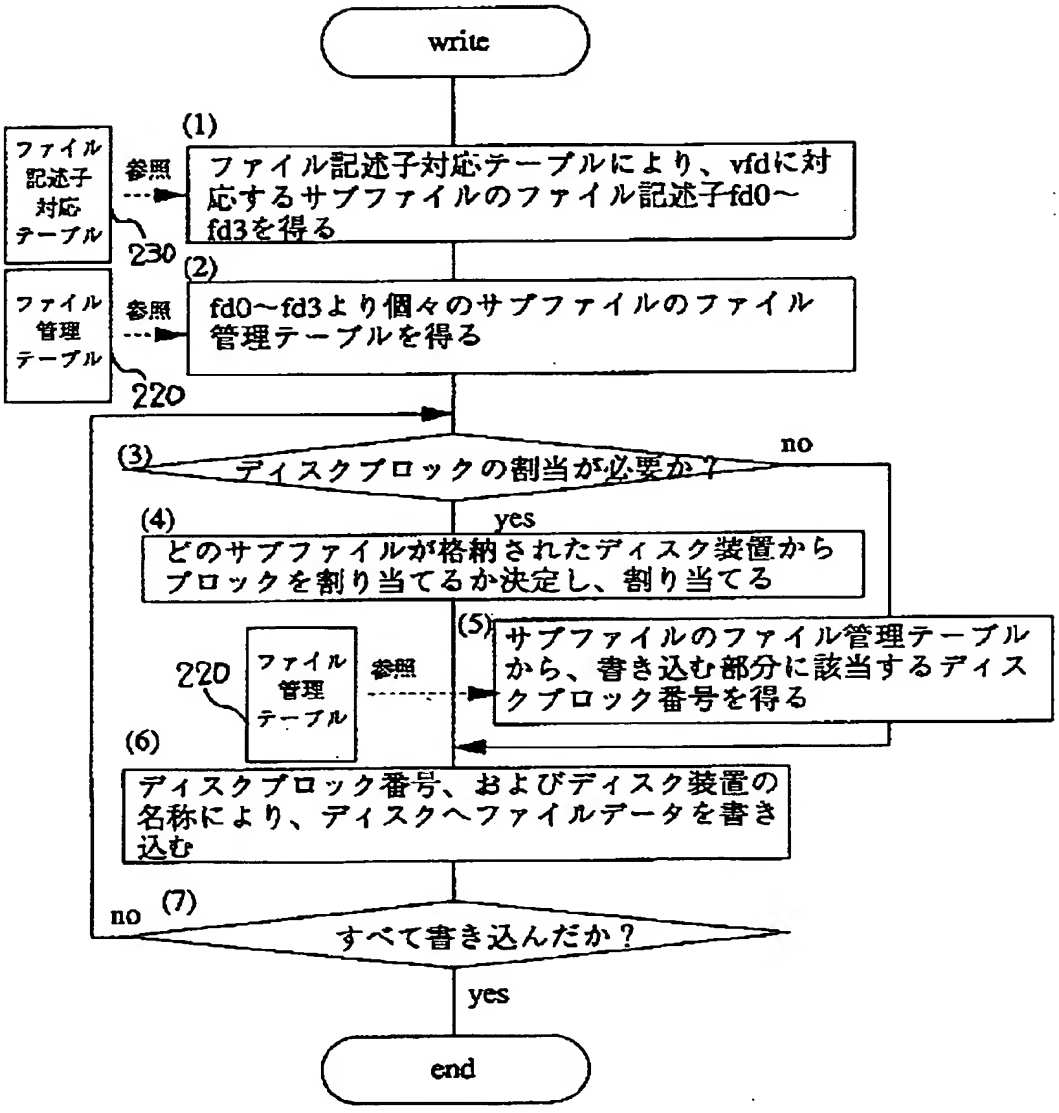
【図２１】

図21



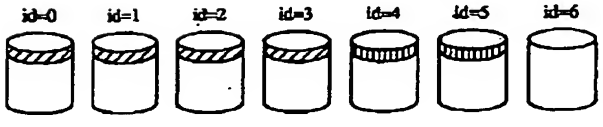
【図１５】

図15



【図２８】

図28

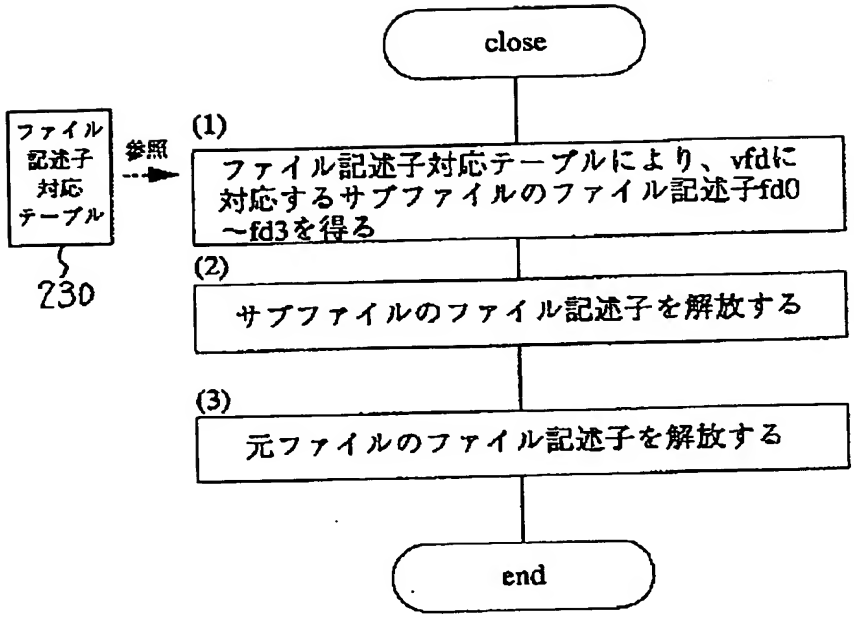


【図３０】

図30

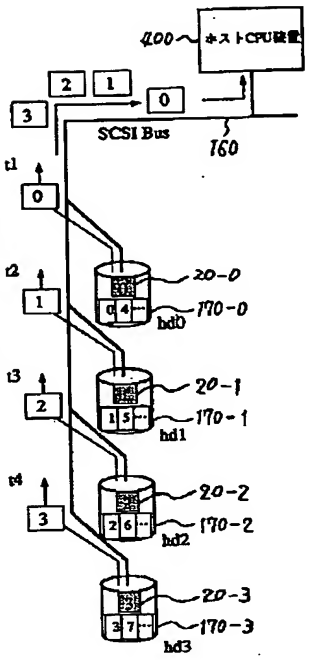
【図１６】

図16



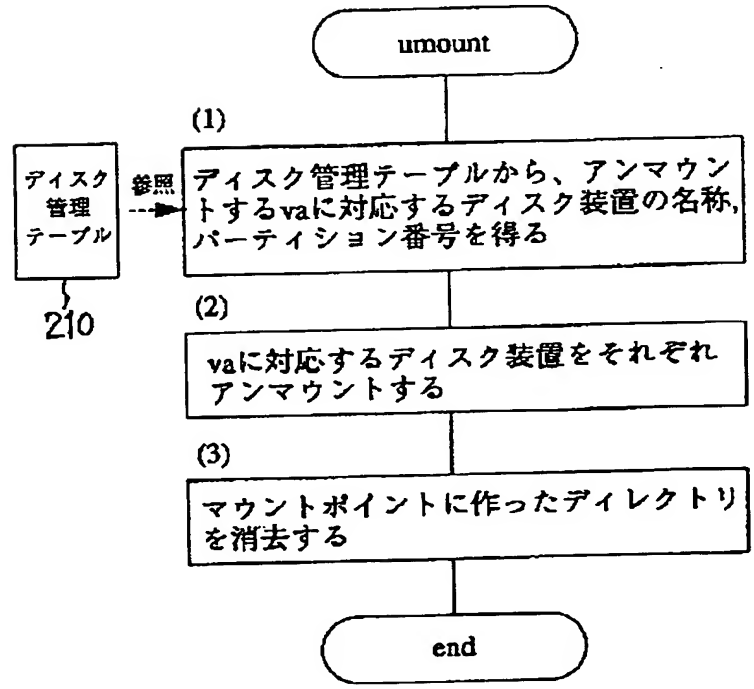
【図４８】

図48



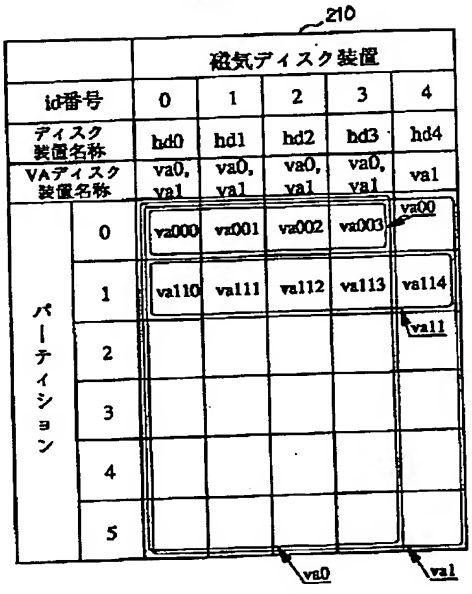
【図１７】

図17



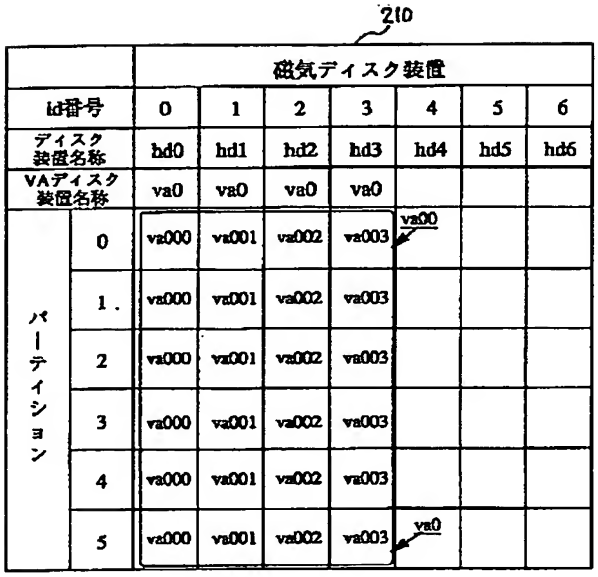
【図４２】

図42



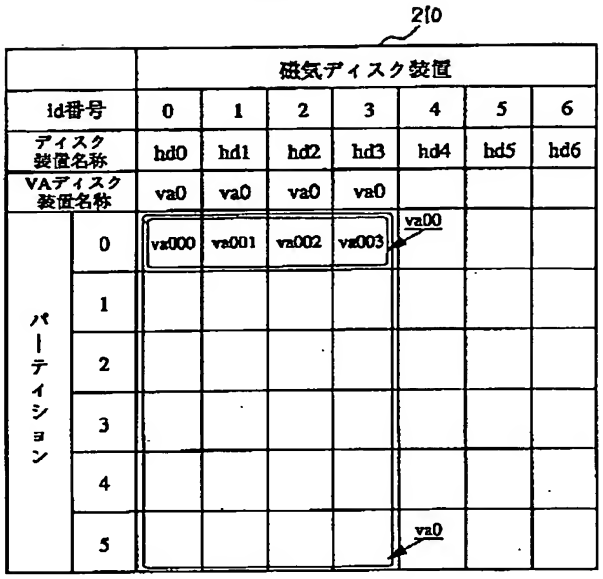
【図２３】

図２３



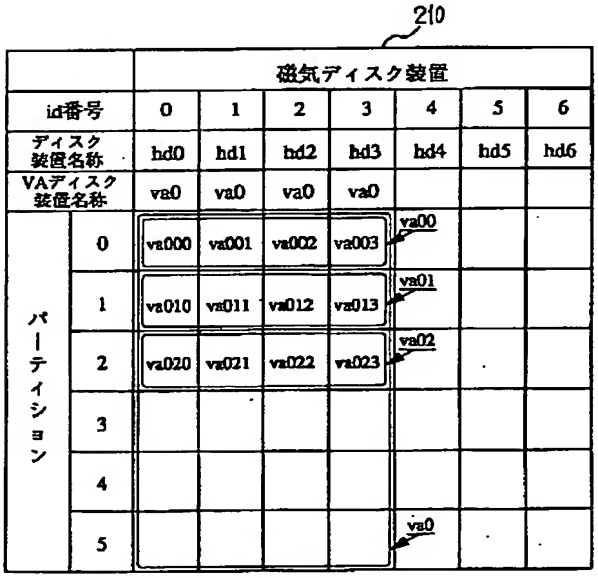
【図２５】

図２５



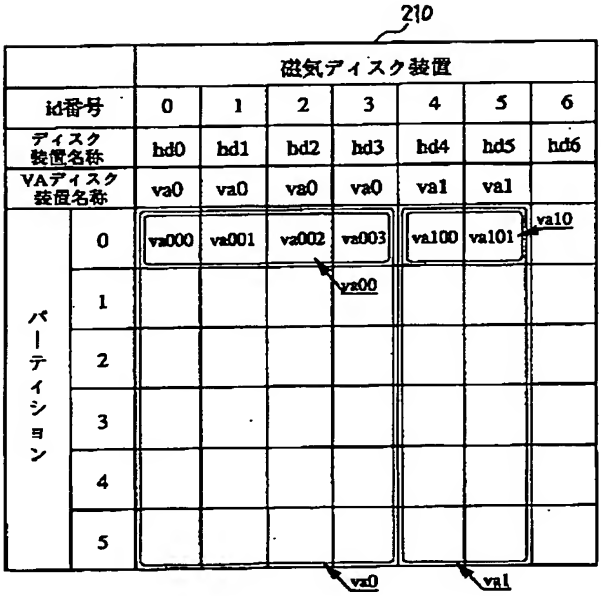
【図２７】

図２７



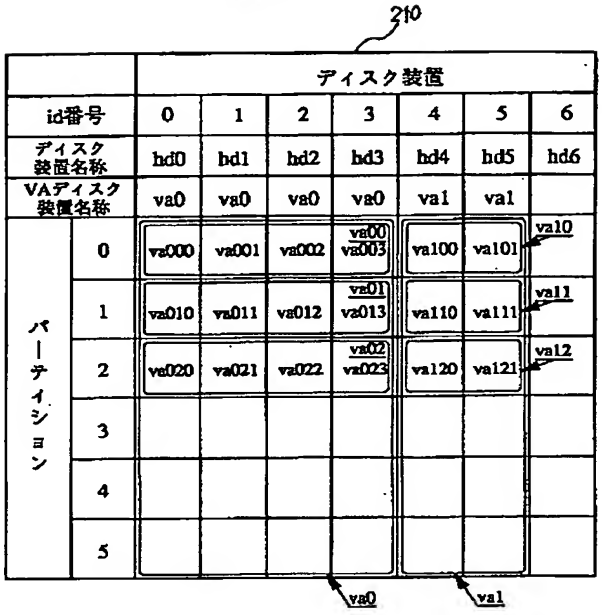
【図２９】

図２９



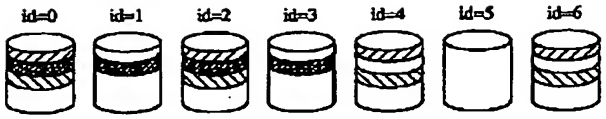
【図３１】

図３１



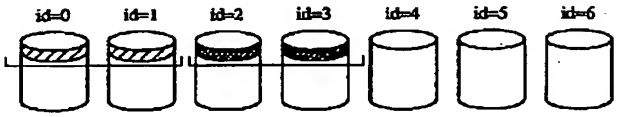
【図３２】

図３２



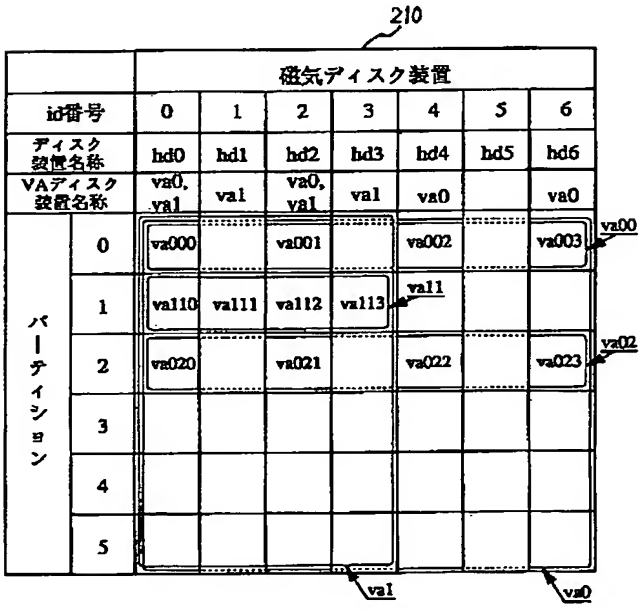
【図３４】

図３４



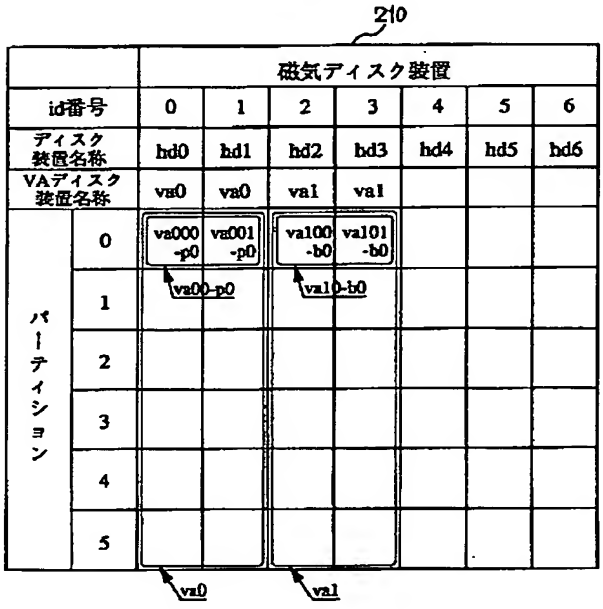
【図３３】

図３３



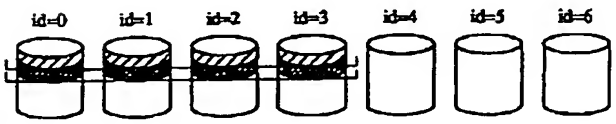
【図３５】

図３５



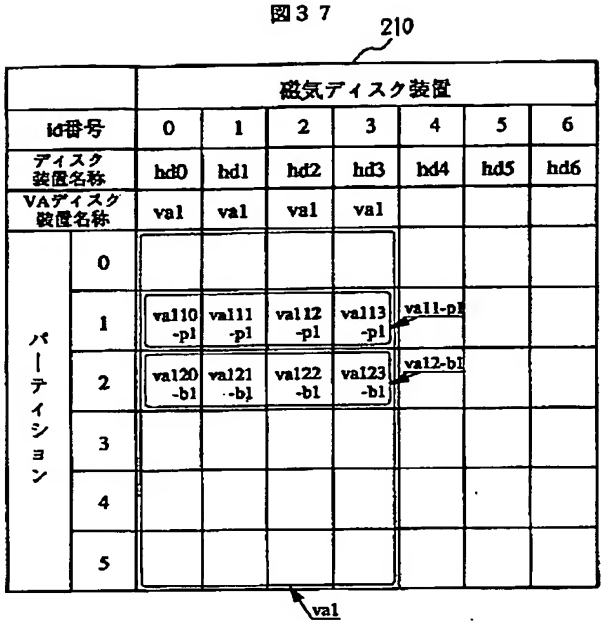
【図３６】

図３６



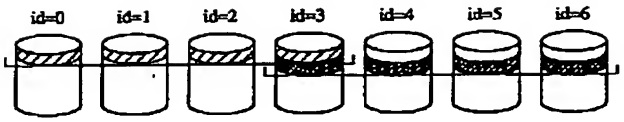
【図３７】

図３７



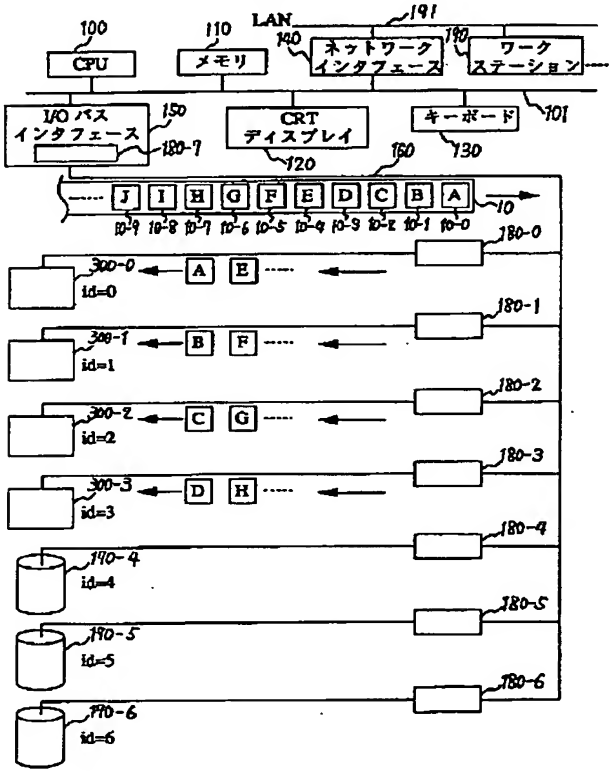
【図３８】

図38



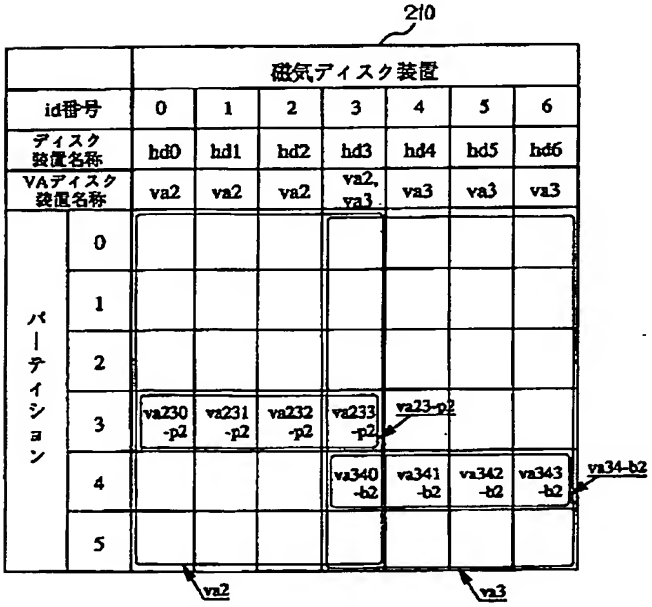
【図４０】

図40



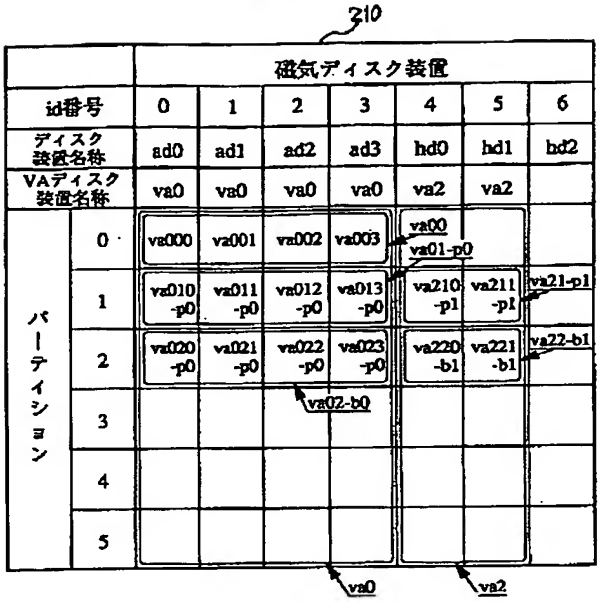
【図３９】

図３９



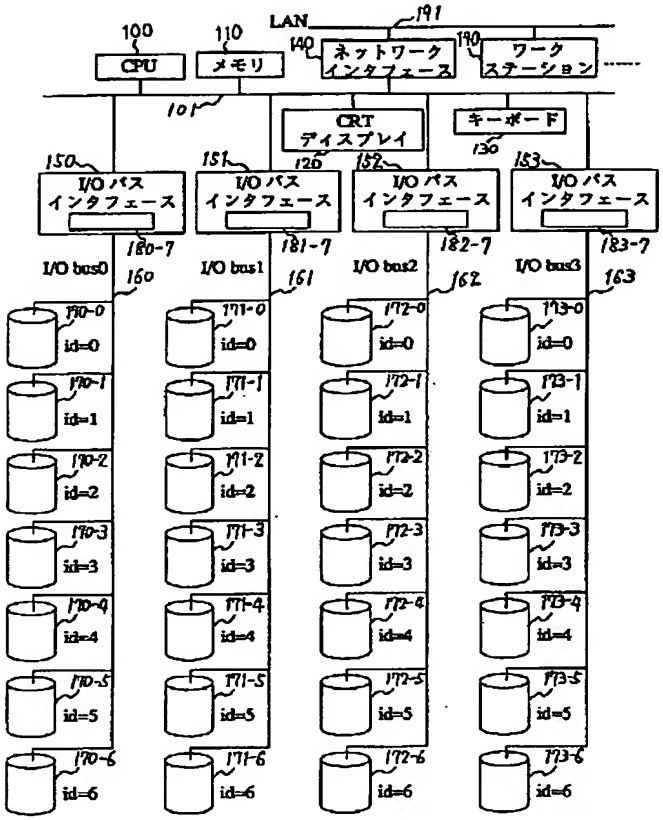
【図４１】

図４１



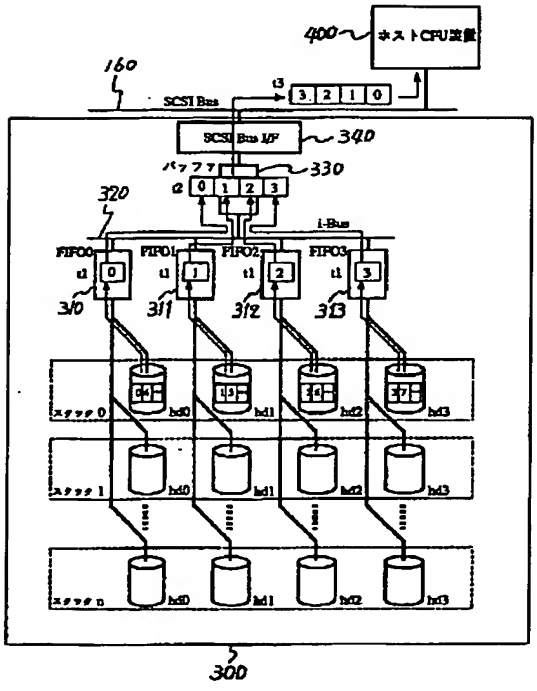
【図４４】

図44



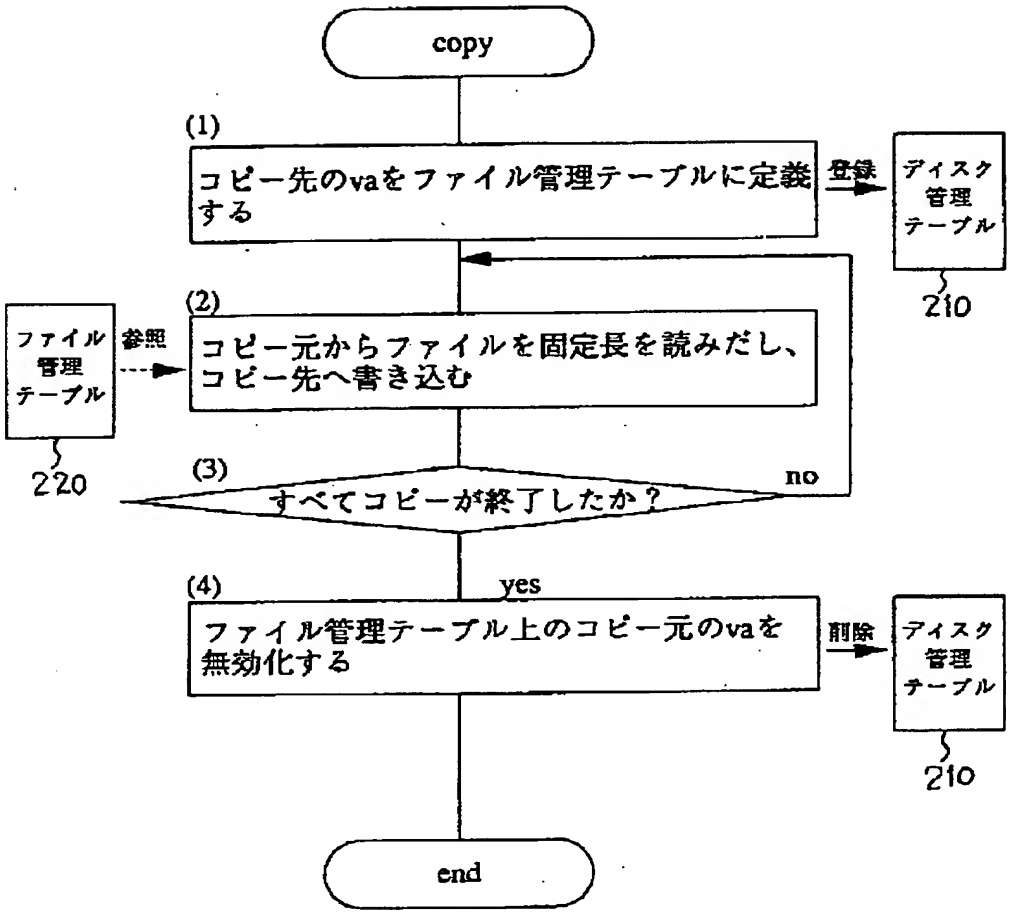
【図４６】

図46



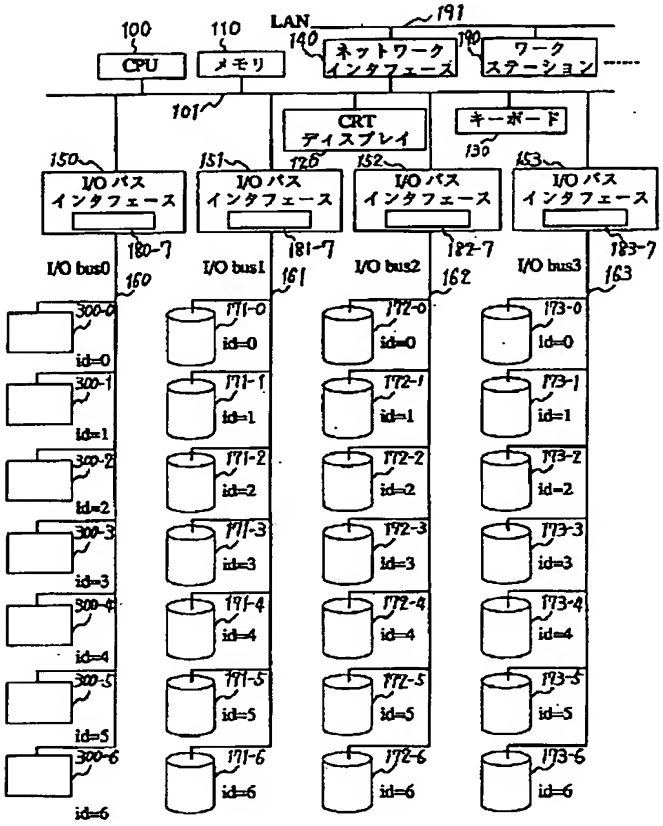
【図４３】

図43


copy
(1)
コピー先のvaをファイル管理テーブルに定義する
登録
ディスク管理テーブル
210
(2)
ファイル管理テーブル
参照
コピー元からファイルを固定長を読みだし、コピー先へ書き込む
220
(3)
すべてコピーが終了したか？
no
yes
(4)
ファイル管理テーブル上のコピー元のvaを無効化する
削除
ディスク管理テーブル
210
end

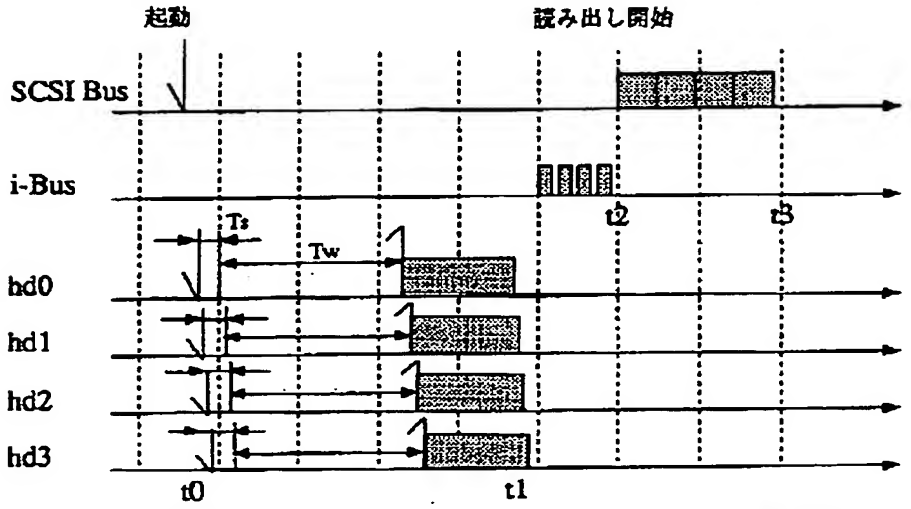
【図４５】

図45



【図４７】

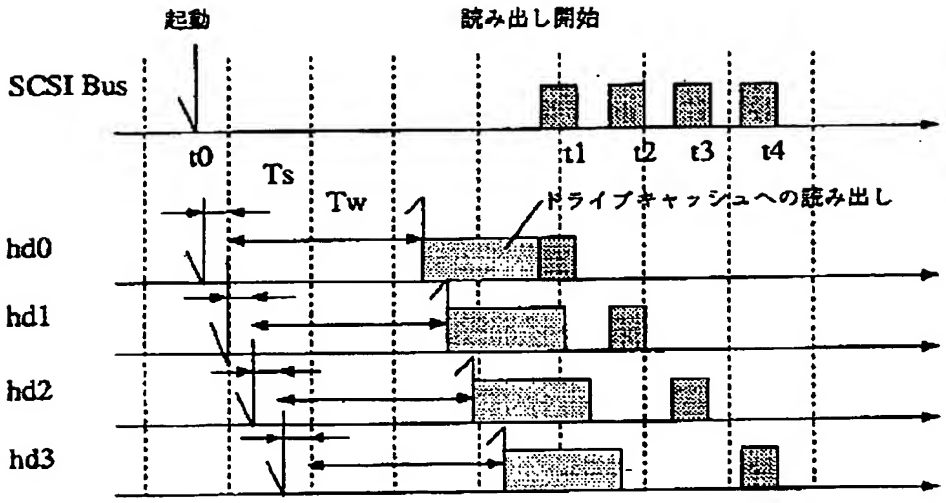
図47



Ts:ソフトウエアオーバヘッド　　Tw:シーク+回転待ち

【図４９】

図49



Ts:ソフトウエアオーバヘッド　　Tw:シーク+回転待ち

フロントページの続き

(72)発明者　加藤　寛次
神奈川県川崎市幸区鹿島田890番地株式会社日立製作所システム開発本部内

(72)発明者　鈴木　広義
神奈川県横浜市戸塚区戸塚町5030番地　株式会社日立製作所ソフトウェア事業部内

(72)発明者　牧　敏行
神奈川県秦野市堀山下１番地　日立コンピュータエンジニアリング株式会社内